

Testing Assignments to Constraint Satisfaction Problems

Hubie Chen
Univ. País Vasco and IKERBASQUE
E-20018 San Sebastián, Spain
hubie.chen@ehu.es

Matt Valeriote*
McMaster University
Hamilton, Canada
matt@math.mcmaster.ca

Yuichi Yoshida†
National Institute of Informatics
Chiyoda-ku, Tokyo 101-8430, Japan
and
Preferred Infrastructure, Inc.
Bunkyo-ku, Tokyo 113-0033, Japan
yyoshida@nii.ac.jp

Abstract

For a finite relational structure \mathbf{A} , let $\text{CSP}(\mathbf{A})$ denote the CSP instances whose constraint relations are taken from \mathbf{A} . The resulting family of problems $\text{CSP}(\mathbf{A})$ has been considered heavily in a variety of computational contexts. In this article, we consider this family from the perspective of property testing: given an instance of a CSP and query access to an assignment, one wants to decide whether the assignment satisfies the instance, or is far from so doing. While previous works on this scenario studied concrete templates or restricted classes of structures, this article presents comprehensive classification theorems.

Our first contribution is a dichotomy theorem completely characterizing the structures \mathbf{A} such that $\text{CSP}(\mathbf{A})$ is constant-query testable:

- If \mathbf{A} has a majority polymorphism and a Maltsev polymorphism, then $\text{CSP}(\mathbf{A})$ is constant-query testable with one-sided error.
- Else, testing $\text{CSP}(\mathbf{A})$ requires a super-constant number of queries.

Let $\exists\text{CSP}(\mathbf{A})$ denote the extension of $\text{CSP}(\mathbf{A})$ to instances which may include existentially quantified variables. Our second contribution is to classify all structures \mathbf{A} in terms of the number of queries needed to test assignments to instances of $\exists\text{CSP}(\mathbf{A})$, with one-sided error. More specifically, we show the following trichotomy:

- If \mathbf{A} has a majority polymorphism and a Maltsev polymorphism, then $\exists\text{CSP}(\mathbf{A})$ is constant-query testable with one-sided error.
- Else, if \mathbf{A} has a $(k+1)$ -ary near-unanimity polymorphism for some $k \geq 2$, and no Maltsev polymorphism then $\exists\text{CSP}(\mathbf{A})$ is not constant-query testable (even with two-sided error) but is sublinear-query testable with one-sided error.
- Else, testing $\exists\text{CSP}(\mathbf{A})$ with one-sided error requires a linear number of queries.

*Supported by a grant from the Natural Sciences and Engineering Research Council of Canada

†Supported by JSPS Grant-in-Aid for Young Scientists (B) (No. 26730009), MEXT Grant-in-Aid for Scientific Research on Innovative Areas (24106003), and JST, ERATO, Kawarabayashi Large Graph Project.

1 Introduction

1.1 Background

In property testing, the goal is to design algorithms that distinguish objects satisfying some predetermined property P from objects that are far from satisfying P . More specifically, for $\epsilon, \delta > 0$, an algorithm is called an (ϵ, δ) -tester for a property P , if given an input I , it accepts with probability at least $1 - \delta$ if the input satisfies P , and it rejects with probability at least $1 - \delta$ if the input I is ϵ -far from satisfying P . Roughly speaking, we say that I is ϵ -far from P if we must modify at least an ϵ -fraction of I to make I satisfy P . When $\delta = 1/3$, we simply call it an ϵ -tester. A tester is called a *one-sided error tester* if it always accepts when I satisfies P . In contrast, a standard tester is sometimes called a *two-sided error tester*. As one motivation of property testing is to design algorithms that run in time sublinear in the input size, we assume query access to the input, and we measure the efficiency of a tester by its *query complexity*. We refer to [18, 27, 28] for surveys on property testing.

In *constraint satisfaction problems* (for short, CSPs), one is given a set of variables and a set of constraints imposed on the variables, and the task is to find an assignment of the variables that satisfies all of the given constraints. By restricting the relations used to specify constraints, it is known that certain restricted versions of the CSP coincide with many fundamental problems such as SAT, graph coloring, and solvability of systems of linear equations. To formally define these restricted versions of the CSP (and hence, these problems), we consider *relational structures* $\mathbf{A} = (A; \Gamma)$, where A is a finite set and Γ consists of a finite set of finitary relations over A . In this context, Γ is sometimes referred to as a *constraint language* over A and \mathbf{A} as a *template*. Then, we define $\text{CSP}(\mathbf{A})$ to be those instances of the CSP whose constraint relations are taken from Γ . In recent years, computational aspects of $\text{CSP}(\mathbf{A})$ have been heavily studied, in the decision setting [21, 10, 3, 5], in counting complexity [11, 15], in computational learning theory [21, 14], and in optimization and approximation [26, 29, 13, 30, 31]. See also the survey by Barto [4] for an overview of this line of research.

In this paper, we consider the problem family $\text{CSP}(\mathbf{A})$ from the perspective of property testing, in particular, we consider the task of testing assignments to CSPs. Relative to a relational structure \mathbf{A} , an input consists of a tuple $(\mathcal{I}, \epsilon, f)$, where \mathcal{I} is an instance of $\text{CSP}(\mathbf{A})$ with weights on the variables, ϵ is an error parameter, and f is an assignment to \mathcal{I} . In the studied model, the tester has full access to \mathcal{I} and query access to f , that is, a variable x can be queried to obtain the value of $f(x)$. In this sense, assignment testing lies in the *massively parameterized model* [25]. We say that f is ϵ -far from satisfying \mathcal{I} if one must modify at least an ϵ -fraction of f (with respect to the weights) to make f a satisfying assignment of \mathcal{I} , and we say that f is ϵ -close otherwise. It is always assumed that \mathcal{I} has a satisfying assignment as otherwise we can immediately reject the input (in this context, one does not care about time complexity). The objective of assignment testing of CSPs is to correctly decide whether f is a satisfying assignment of \mathcal{I} or is ϵ -far from being so with probability at least $2/3$. When f does not satisfy \mathcal{I} but is ϵ -close to satisfying \mathcal{I} , we can output anything.

In assignment testing, we say that the query complexity of a tester is constant/sublinear/linear if it is constant/sublinear/linear in the number of variables of an instance. The main problem addressed in this paper is to reveal the relationship between a relational structure \mathbf{A} and the number of queries needed to test $\text{CSP}(\mathbf{A})$ and a related problem class $\exists\text{CSP}(\mathbf{A})$.

1.2 Contributions

While previous works on testing assignments to the problems $\text{CSP}(\mathbf{A})$ studied concrete templates \mathbf{A} or restricted classes of structures, this article presents comprehensive classification theorems.

The first contribution of this paper is a dichotomy theorem that *completely characterizes* the constant-query testable CSPs. Before describing our characterization, we introduce the algebraic notion of a polymorphism which is key to the description and obtention of our results. Let R be an r -ary relation on a set A . A $(k$ -ary) operation $f : A^k \rightarrow A$ is said to be a *polymorphism* of R (or R is *preserved* by f) if for any set of k r -tuples $(a_1^1, \dots, a_r^1), (a_1^2, \dots, a_r^2), \dots, (a_1^k, \dots, a_r^k) \in R$, the tuple $(f(a_1^1, \dots, a_1^k), \dots, f(a_r^1, \dots, a_r^k))$ also belongs to R . An operation f is a *polymorphism* of a relational structure \mathbf{A} if it is a polymorphism of each of its relation. We define the *algebra* of \mathbf{A} , denoted by $\text{Alg}(\mathbf{A})$, to be the pair $(A; \text{Pol}(\mathbf{A}))$, where $\text{Pol}(\mathbf{A})$ is the set of all polymorphisms of \mathbf{A} .

Definition 1.1. Let A be a nonempty set. A majority operation on A is a ternary operation $m : A^3 \rightarrow A$ such that $m(b, a, a) = m(a, b, a) = m(a, a, b) = a$ for all $a, b \in A$. A Maltsev operation on A is a ternary operation $p : A^3 \rightarrow A$ such that $p(b, a, a) = p(a, a, b) = b$ for all $a, b \in A$. For $k \geq 2$, an operation $n : A^{(k+1)} \rightarrow A$ is a $(k+1)$ -ary near unanimity operation on A if for all $a, b \in A$,

$$n(b, a, a, \dots, a) = n(a, b, a, \dots, a) = \dots = n(a, a, \dots, a, b) = a.$$

(Note that a majority operation is a 3-ary near-unanimity operation.)

Theorem 1.2. Let \mathbf{A} be a relational structure. The following dichotomy holds.

- (1) If \mathbf{A} has a majority polymorphism and a Maltsev polymorphism, then $\text{CSP}(\mathbf{A})$ is constant-query testable (with one-sided error).
- (2) Else, testing $\text{CSP}(\mathbf{A})$ requires a super-constant number of queries.

This theorem generalizes characterizations of constant-query testable List H -homomorphisms [33] and Boolean CSPs [7] to general CSPs. In Section 3 we will describe the particularly nice structure of relations over templates that have majority and Maltsev polymorphisms and use this to prove the theorem. For the moment, let us consider a number of example templates to which the positive result of this theorem applies.

Example 1.3. The template \mathbf{A} over the Boolean domain $\{0, 1\}$ whose only relation is \neq has both majority and Maltsev polymorphisms. Note that $\text{CSP}(\mathbf{A})$ coincides with the graph 2-coloring problem.

More generally, the template \mathbf{A} over a finite domain where each relation is a bijection on A has both majority and Maltsev polymorphisms, and instances of $\text{CSP}(\mathbf{A})$ for such templates \mathbf{A} coincide with instances of the problem which is the subject of the *unique games conjecture* [23]. \square

Example 1.4. Another class of finite structures that have both majority and Maltsev polymorphisms are those that have a *discriminator* operation as a polymorphism. On a set A the discriminator operation $d_A(x, y, z)$ is the operation such that if $x = y$ then $d(x, y, z) = z$ and if $x \neq y$, $d(x, y, z) = x$. From this definition, it is immediate that d_A is a Maltsev operation on A , and that $d(x, d(x, y, z), z)$ is a majority operation on A . Any finite product of finite fields will have a discriminator term operation ([12]) and so any finite relational structure whose relations are compatible with the operations of such a ring will have majority and Maltsev polymorphism. \square

Example 1.5. For p a prime number, let \mathbb{F}_p be the field of size p , and let \mathbb{R} be the ring $\mathbb{F}_2 \times \mathbb{F}_3 \times \mathbb{F}_5$. Then as noted in Example 1.4, \mathbb{R} has a discriminator term operation. Let \mathbf{R} be the structure with domain R and set of relations Γ consisting of intersections of the following binary relations on R : For $p = 2, 3$, or 5 ,

- $C_p = \{((a_2, a_3, a_5), (b_2, b_3, b_5)) \mid a_p = b_p\}$,
- For $a \in \mathbb{F}_p$, $E_a = \{((a_2, a_3, a_5), (b_2, b_3, b_5)) \mid a_p = a\}$,
- For $b \in \mathbb{F}_p$, $E_b = \{((a_2, a_3, a_5), (b_2, b_3, b_5)) \mid b_p = b\}$,

So relations in Γ can express that pairs of elements in R are congruent modulo 2, 3, or 5 in the corresponding coordinate and/or that a certain coordinate is equal to some fixed value. These relations are invariant under the discriminator term operation of \mathbb{R} and so according to Theorem 1.2, $\text{CSP}(\mathbf{R})$ has constant query complexity. \square

Examples of structures that satisfy the first condition of Theorem 1.2 but that do not have a discriminator operation as a polymorphism can be derived from finite Heyting algebras.

Example 1.6. Consider the five-element Heyting algebra \mathbb{M} presented in [20, Figure 1]. (Heyting algebras are bounded distributive lattices that also have a binary “implication” operation; they serve as algebraic models of propositional intuitionistic logic.) This algebra has universe $M = \{0, a, b, e, 1\}$; the two equivalence relations α and β that partition M into blocks $\{\{0, a\}, \{b, e, 1\}\}$ and $\{\{0, b\}, \{a, e, 1\}\}$ (respectively) are preserved by the operations of the algebra. Since \mathbb{M} has majority and Maltsev term operations (the operations $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ and $(x \rightarrow y) \rightarrow z \wedge (z \rightarrow y) \rightarrow x$ respectively), then the structure $\mathbf{M} = (M; \alpha, \beta)$ has majority and Maltsev polymorphisms. The only other non-trivial binary relation on M that is definable by a primitive-positive formula over \mathbf{M} is $\alpha \cap \beta$. \square

Example 1.7. Bulatov and Marx provide yet another example of a structure having both a majority and a Maltsev polymorphism, in [9, Example 1.1]. \square

We next consider existentially quantified CSPs ($\exists\text{CSP}$ s for short). The difference between CSPs and $\exists\text{CSP}$ s is that, in an instance of $\exists\text{CSP}$, existentially quantified variables may appear. So, an instance of $\exists\text{CSP}$ may be defined as a primitive positive formula (pp-formula) over a relational structure. Primitive positive formulas are known as *conjunctive queries* in the database theory literature; they are arguably the most heavily studied class of database queries, and the problem $\exists\text{CSP}$ can be associated with the problem of *conjunctive query evaluation*.

For a relational structure $\mathbf{A} = (A; \Gamma)$, we define $\exists\text{CSP}(\mathbf{A})$ to be the collection of instances of $\exists\text{CSP}$ whose constraint relations are taken from Γ . Our second contribution is to provide a complete classification of all structures \mathbf{A} in terms of the number of queries needed to test assignments of instances of $\exists\text{CSP}(\mathbf{A})$ with one-sided error:

Theorem 1.8. *Let \mathbf{A} be a relational structure. Then, the following trichotomy holds.*

- (1) *If \mathbf{A} has a majority polymorphism and a Maltsev polymorphism, then $\exists\text{CSP}(\mathbf{A})$ is constant-query testable with one-sided error.*

- (2) Else, if \mathbf{A} has a $(k+1)$ -ary near-unanimity polymorphism for some $k \geq 2$, and no Maltsev polymorphism then $\exists\text{CSP}(\mathbf{A})$ is not constant-query testable (even with two-sided error) but is sublinear-query testable with one-sided error.¹
- (3) Else, testing $\exists\text{CSP}(\mathbf{A})$ with one-sided error requires a linear number of queries.

Let us point out that the problem families $\text{CSP}(\mathbf{A})$ and $\exists\text{CSP}(\mathbf{A})$ exhibit the same dichotomy for constant-query testability, and in particular the positive result there is robust with respect to the introduction of quantifiers. An implication of Theorem 1.8 is this: if the dichotomy for sublinear-query testability for $\text{CSP}(\mathbf{A})$ was not the same as that for $\exists\text{CSP}(\mathbf{A})$, then the positive result for that dichotomy would not enjoy this robustness property, and hence such a positive result would have to crucially exploit the absence of quantifiers. Hence Theorem 1.8 reveals information about the form of a potential trichotomy for $\text{CSP}(\mathbf{A})$.

A special feature of templates \mathbf{A} that have a $(k+1)$ -ary near-unanimity polymorphism is that any relation that is definable by a pp-formula over \mathbf{A} can be decomposed into a number of k -ary relations that are also pp-definable over \mathbf{A} .

Example 1.9. Consider the relational structure \mathbf{A} over the Boolean domain $\{0,1\}$ whose only relation is \leq . This structure is readily verified to have a majority polymorphism (note that over the Boolean domain, there is indeed a unique majority operation), and does not have a Maltsev polymorphism: for any Maltsev operation p , it holds that applying p to the tuples $(1,1), (0,1), (0,0)$, which are in the relation \leq , yields $(1,0)$, which are not in the relation \leq . Thus, Theorem 1.8 implies that $\exists\text{CSP}(\mathbf{A})$ is not constant-query testable but is sublinear-query testable with one-sided error. \square

Example 1.10. We can generalize the previous example as follows. Let D be any finite set of size greater than or equal to 2, and consider the *dual discriminator operation* Δ defined as follows: $\Delta(x,y,z)$ is equal to x if $x = y$, and is equal to z otherwise. Consider the relational structure \mathbf{A} with universe D and the following relations: each unary relation; each graph of a permutation (on D); and, each *two-fan relation*. Here, a *two-fan relation* is a binary relation $R \subseteq D \times D$ such that there exist elements $a, b \in D$ where $R = (\{a\} \times \pi_2(R)) \cup (\pi_1(R) \times \{b\})$. It is straightforward to verify that Δ is a majority polymorphism of \mathbf{A} . On the other hand, let $a, b \in D$ be arbitrary elements, and consider the relation $S = (\{a\} \times D) \cup (D \times \{b\})$. The relation S is a two-fan relation and so is a relation of \mathbf{A} , but does not have a Maltsev polymorphism; we argue this as follows. Let a' be an element of D distinct from a , and let b' be an element of D distinct from b . We have that the tuples $(a',b), (a,b), (a,b')$ are in S , but if we apply any Maltsev polymorphism p to them, we obtain (a',b') which is not in S . The structure \mathbf{A} thus does not have a Maltsev polymorphism; we obtain by Theorem 1.8 that $\exists\text{CSP}(\mathbf{A})$ is not constant-query testable but is sublinear-query testable with one-sided error. \square

1.3 Proof outline

We now describe outlines of our proofs of Theorems 1.2 and 1.8.

¹ We remark that the combination of having a $(k+1)$ -ary near unanimity polymorphism for some $k \geq 2$ and a Maltsev polymorphism is equivalent to having majority and Maltsev polymorphisms [12].

A has majority and Maltsev polymorphisms $\Rightarrow \exists\text{CSP}(\mathbf{A})$ is constant-query testable. We first look at (1) of Theorem 1.2 and 1.8. As $\exists\text{CSPs}$ are a generalization of CSPs, it suffices to consider $\exists\text{CSPs}$. Let $(\mathcal{I}, \epsilon, f)$ be an input of the assignment testing of $\exists\text{CSP}(\mathbf{A})$. First, we preprocess \mathcal{I} so that it becomes 2-consistent (see Section 2 for the formal definition). Using the 2-consistency of \mathcal{I} and the majority polymorphism of \mathbf{A} we can assume that for each variable x of \mathcal{I} , the set of allowed values for x forms a domain A_x that is the universe of an algebra \mathbb{A}_x that is a factor (i.e., a homomorphic image of a subalgebra) of $\text{Alg}(\mathbf{A})$, the algebra of polymorphisms of \mathbf{A} . Also, we can assume that for each pair of variables x, y of \mathcal{I} there is a unique binary constraint of \mathcal{I} with scope (x, y) and constraint relation R_{xy} , with R_{xy} the universe of some subalgebra of $\mathbb{A}_x \times \mathbb{A}_y$. Furthermore these are the only constraints of \mathcal{I} .

In order to test whether f satisfies \mathcal{I} , we use three types of reductions: a factoring reduction, a splitting reduction, and an isomorphism reduction. Each reduction produces an instance \mathcal{I}' and an assignment f' such that f' satisfies \mathcal{I}' if f satisfies \mathcal{I} , and f' is $\Omega(\epsilon)$ -far from satisfying \mathcal{I}' if f is ϵ -far from satisfying \mathcal{I} . For simplicity, we focus on how we create a new instance \mathcal{I}' here.

The objective of the factoring reduction is to factor, for each variable x of \mathcal{I} , the domain A_x by any congruence θ of \mathbb{A}_x (i.e., an equivalence relation on A_x that is compatible with the operations of \mathbb{A}_x) for which none of the constraint relations of \mathcal{I} distinguish between θ -related values of A_x .

After ensuring that all of the domains A_x of \mathcal{I} cannot be factored, we then employ a splitting reduction to ensure that for each variable x of \mathcal{I} the algebra \mathbb{A}_x is subdirectly irreducible, i.e., cannot be represented as a subdirect product of non-trivial algebras. For any variable x for which \mathbb{A}_x can be represented as a subdirect product of non-trivial algebras \mathbb{A}_x^1 and \mathbb{A}_x^2 we replace the variable x by the new variables x_1 and x_2 and the domain A_x by the domains A_x^1 and A_x^2 . For any other variable y of \mathcal{I} , we “split” the constraint relation R_{yx} (and its inverse R_{xy}) into two relations R_{yx_1} and R_{yx_2} that are together equivalent to the original one. We then add these two new relations (and their inverses) to \mathcal{I} , along with A_x , now regarded as a binary relation from the variable x_1 to x_2 .

After performing the splitting reduction and the factoring reduction, we next define a binary relation \sim on the set of variables of \mathcal{I} such that $x \sim y$ if and only if the constraint relation R_{xy} is the graph of an isomorphism from \mathbb{A}_x to \mathbb{A}_y . Using 2-consistency and the fact that the domains of \mathcal{I} are subdirectly irreducible and cannot be factored, it follows that, unless \mathcal{I} is trivial, the relation \sim will be a non-trivial equivalence relation. Within each \sim -class, the domains are isomorphic via the corresponding constraint relations of \mathcal{I} , and this allows us to produce an isomorphism-reduced instance \mathcal{I}' by restricting \mathcal{I} to a set of variables representing each of the \sim -classes.

After performing this isomorphism reduction, the resulting instance may have domains which can be further factored, allowing us to apply the factoring reduction to produce a smaller instance. We show that if we reach a point at which none of the three reductions can be applied, the instance must be trivial, either having just a single variable, or for which $|A_x| = 1$ for all variables x . We also show that this point will be reached after applying the reductions at most $|A|$ -times.

In Section 3, we will see how these reductions work on the template in Example 1.5.

CSP(A) is constant-query testable $\Rightarrow \mathbf{A}$ has majority and Maltsev polymorphisms. Now we look at (2) of Theorem 1.2 and the hardness part of (2) of Theorem 1.8. As $\exists\text{CSPs}$ are a generalization of CSPs, it suffices to consider CSPs. We show that if \mathbf{A} does not have these two types of polymorphisms, then we cannot test $\text{CSP}(\mathbf{A})$ with a constant number of queries. We use that having these two types of polymorphisms is equivalent to \mathbf{A} having a Maltsev polymorphism

and that the variety of algebras generated by $\text{Alg}(\mathbf{A})$ is congruence meet semidistributive [19]. The paper [24] provides a characterization of this condition in terms of the existence of two special polymorphisms of \mathbf{A} . When the variety generated by $\text{Alg}(\mathbf{A})$ is not congruence meet semidistributive, then it can be easily shown from [7, 33] that testing $\text{CSP}(\mathbf{A})$ requires a linear number of queries. When \mathbf{A} does not have a Maltsev polymorphism, then we can reduce $\text{CSP}(\mathbf{A}')$ to $\text{CSP}(\mathbf{A})$, where the structure \mathbf{A}' has a binary non-rectangular relation. Then, by replacing the 2-SAT relations with this binary non-rectangular relation, we can reuse the argument for showing a super-constant lower bound for 2-SAT in [17] to obtain a super-constant lower bound for $\text{CSP}(\mathbf{A})$.

A has a $(k + 1)$ -ary near-unanimity polymorphism, for some $k \geq 2 \Rightarrow \exists \text{CSP}(\mathbf{A})$ is sublinear-query testable. Now we consider the testability part of (2) of Theorem 1.8. It is known that, if \mathbf{A} has a $(k + 1)$ -ary near unanimity polymorphism, then $\text{CSP}(\mathbf{A})$ is sublinear-query testable with one-sided error in the unweighted case [7]. We slightly modify their argument so that we can handle weights.

$\exists \text{CSP}(\mathbf{A})$ is sublinear-query testable with one-sided error $\Rightarrow \mathbf{A}$ has a $(k + 1)$ -ary near unanimity polymorphism, for some $k \geq 2$. Finally, we consider (3) of Theorem 1.8. We use that \mathbf{A} has a $(k + 1)$ -ary near unanimity polymorphism for some $k \geq 2$ if and only if the variety of algebras generated by $\text{Alg}(\mathbf{A})$ is congruence meet semidistributive and congruence modular [19, 2]. We already mentioned that if this variety is not congruence meet semidistributive then $\exists \text{CSP}(\mathbf{A})$ is not sublinear-query testable (even with two-sided error). To complete the argument we show that if the variety is not congruence modular then, by building on ideas developed in [14], we can reduce the problem of testing assignments of a circuit in monotone NC^1 to $\exists \text{CSP}(\mathbf{A})$. Note that majority functions are in monotone NC^1 [32], and we can easily show a linear lower bound for one-sided error testers that test assignments of majority functions. Hence, we get a linear lower bound for $\exists \text{CSP}(\mathbf{A})$.

1.4 Related work

Assignment testing of CSPs was implicitly initiated by [17]. There, it was shown that 2-CSPs are testable with $O(\sqrt{n})$ queries and require $\Omega(\log n / \log \log n)$ queries for any fixed $\epsilon > 0$. On the other hand, 3-SAT [6], 3-LIN [6], and Horn SAT [7] require $\Omega(n)$ queries to test.

The universal algebraic approach was first used in [33] to study the assignment testing of the list H -homomorphism problem. For graphs G, H , and list constraints $L_v \subseteq V(H)$ ($v \in V(G)$), we say that a mapping $f : V(G) \rightarrow V(H)$ is a *list homomorphism* from G to H with respect to the list constraints L_v ($v \in V(G)$) if $f(v) \in L_v$ for any $v \in V(G)$ and $(f(u), f(v)) \in E(H)$ for any $(u, v) \in E(G)$. Then, the corresponding assignment testing problem, parameterized by a graph H , is the following: The input is a tuple $(G, \{L_v\}_{v \in V(H)}, f, \epsilon)$, where G is a (weighted) graph, $L_v \subseteq V(H)$ ($v \in V(G)$) are list constraints, $f : V(G) \rightarrow V(H)$ is a mapping given as a query access, and ϵ is an error parameter. The goal is testing whether f is a list H -homomorphism from G or ϵ -far from being so, where ϵ -farness is defined analogously to testing assignments of CSPs. It was shown in [33] that the algebra (or the variety) associated with the list H -homomorphism characterizes the query complexity, and that list H -homomorphism is constant-query (resp., sublinear-query) testable if and only if H is a reflexive complete graph or an irreflexive complete bipartite graph (resp., a bi-arc graph).

Testing assignments of Boolean CSPs was studied in [7], and in that paper relational structures were classified into three categories: (i) structures \mathbf{A} for which $\text{CSP}(\mathbf{A})$ is constant-query testable, (ii) structures \mathbf{A} for which $\text{CSP}(\mathbf{A})$ is not constant-query testable but sublinear-query testable, and (iii) structures \mathbf{A} for which $\text{CSP}(\mathbf{A})$ is not sublinear-query testable. They also relied on the fact that algebras (or varieties) can be used to characterize query complexity.

1.5 Open problems

Theorem 1.2 characterizes relational structures \mathbf{A} on general domains for which $\text{CSP}(\mathbf{A})$ is constant-query testable. Obtaining a characterization for the sublinear-query testable case is a tantalizing open problem. The main obstacle of this is that we obtained (3) of Theorem 1.8 by reducing the problem of testing assignments of monotone circuits to $\exists\text{CSPs}$. If we do not allow existentially quantified variables, then the number of variables blows up polynomially in the reduction, and a linear lower bound for monotone circuits does not imply a linear lower bound for CSPs.

Theorem 1.8 provides a trichotomy for $\exists\text{CSPs}$ in terms of the number of queries needed to test with one-sided error. Obtaining a similar trichotomy for two-sided error testers is also an interesting open problem. Again the obstacle is that we reduce from the problem of testing assignments of monotone circuits. It is not clear whether this problem is hard even for two-sided error testers.

1.6 Organization

Section 2 introduces the basic notions used throughout this paper. We show the constant-query testability of $\exists\text{CSPs}$ with majority and Maltsev polymorphisms in Section 3. Super-constant lower bounds of CSPs without majority or Maltsev polymorphisms is discussed in Section 4. We give a sublinear-query tester for $\exists\text{CSPs}$ having a $(k+1)$ -ary near unanimity polymorphism, for some $k \geq 2$, in Section 5. In Section 6, we show that, when there is no $(k+1)$ -ary near unanimity polymorphism for any $k \geq 2$, testing $\exists\text{CSPs}$ with one-sided error requires a linear number of queries.

2 Preliminaries

For an integer k , let $[k]$ denote the set $\{1, \dots, k\}$.

Constraint satisfaction problems For an integer $k \geq 1$, a k -ary relation on a domain A is a subset of A^k . A *constraint language* on a domain A is a finite set of relations on A . A (*finite*) *relational structure*, or simply a *structure* $\mathbf{A} = (A; \Gamma)$ consists of a non-empty set A and a constraint language Γ on A .

For a structure $\mathbf{A} = (A; \Gamma)$, we define the problem $\text{CSP}(\mathbf{A})$ as follows. An instance $\mathcal{I} = (V, A, \mathcal{C}, \mathbf{w})$ consists of a set of variables V , a set of constraints \mathcal{C} , and a weight function \mathbf{w} with $\sum_{x \in V} \mathbf{w}(x) = 1$. Here, each constraint $C \in \mathcal{C}$ is of the form $\langle (x_1, \dots, x_k), R \rangle$, where $x_1, \dots, x_k \in V$ are variables, R is a relation in Γ and k is the arity of R . An *assignment* for \mathcal{I} is a mapping $f : V \rightarrow A$, and we say that f is a *satisfying assignment* if f satisfies all the constraints, that is, $(f(x_1), \dots, f(x_k)) \in R$ for every constraint $\langle (x_1, \dots, x_k), R \rangle \in \mathcal{C}$.

For a structure $\mathbf{A} = (A; \Gamma)$, we define the problem $\exists\text{CSP}(\mathbf{A})$ as follows. An instance $\mathcal{I} = (V, V^\exists, \mathcal{C}, \mathbf{w})$ consists of a set of free variables V , a set of existentially quantified variables V^\exists , a set of constraints \mathcal{C} , and a weight function \mathbf{w} with $\sum_{x \in V} \mathbf{w}(x) = 1$. Constraints are imposed on $V \cup V^\exists$. An *assignment* for \mathcal{I} is a mapping $f : V \rightarrow A$, and we say that f is a *satisfying assignment*

if there exists an extension $f' : V \cup V^\exists \rightarrow A$ of f such that $(f'(x_1), \dots, f'(x_k)) \in R$ for every constraint $\langle (x_1, \dots, x_k), R \rangle \in \mathcal{C}$.

Algebras and Varieties: Let $\mathbb{A} = (A; F)$ be an algebra. A set $B \subseteq A$ is a *subuniverse* of \mathbb{A} if for every operation $f \in F$ restricted to B has image contained in B . For a nonempty subuniverse B of an algebra \mathbb{A} , $f|_B$ is the restriction of f to B . The algebra $\mathbb{B} = (B, F|_B)$, where $F|_B = \{f|_B \mid f \in F\}$ is a *subalgebra* of \mathbb{A} . Algebras \mathbb{A}, \mathbb{B} are of the *same type* if they have the same number of operations and corresponding operations have the same arities. Given algebras \mathbb{A}, \mathbb{B} of the same type, the *product* $\mathbb{A} \times \mathbb{B}$ is the algebra with the same type as \mathbb{A} and \mathbb{B} with universe $A \times B$ and operations computed coordinate-wise. A subalgebra \mathbb{C} of $\mathbb{A} \times \mathbb{B}$ is a *subdirect product* of \mathbb{A} and \mathbb{B} if the projections of \mathbb{C} to A and \mathbb{C} to B are both onto. An equivalence relation θ on A is called a *congruence* of an algebra \mathbb{A} if θ is a subalgebra of $\mathbb{A} \times \mathbb{A}$. The collection of congruences of an algebra naturally forms a lattice under the inclusion ordering, and this lattice is called the *congruence lattice* of the algebra. Given a congruence θ on A , we can form the *homomorphic image* \mathbb{A}/θ , whose elements are the equivalence classes of \mathbb{A} and the operations are defined so that the natural mapping from \mathbb{A} to \mathbb{A}/θ is a homomorphism. An operation $f(x_1, \dots, x_n)$ on a set A is *idempotent* if $f(a, a, \dots, a) = a$ for all $a \in A$, an algebra \mathbb{A} is *idempotent* if each of its operations is, and a class of algebras is idempotent if each of its members is. We note that if \mathbb{A} is *idempotent*, then for any congruence θ of \mathbb{A} , the θ -classes are all subuniverses of \mathbb{A} .

A *variety* is a class of algebras of the same type closed under the formation of homomorphic images, subalgebras, and products. For any algebra \mathbb{A} , there is a smallest variety containing \mathbb{A} , denoted by $\mathcal{V}(\mathbb{A})$ and called the *variety generated* by \mathbb{A} . It is well known that any variety is generated by an algebra and that any member of $\mathcal{V}(\mathbb{A})$ is a homomorphic image of a subalgebra of a power of \mathbb{A} .

Many important properties of the algebras in a variety can be correlated with properties of the congruence lattices of its member algebras. In this work we consider several congruence lattice conditions for varieties, including congruence modularity, congruence distributivity, congruence meet semidistributivity, and congruence permutability. Details of these conditions can be found in [19] and more details on the basics of algebras and varieties can be found in [12].

2.1 Assignment problems

An *assignment problem* consists of a set of *instances*, where each instance \mathcal{I} has associated with it a set of variables V , a domain A_v for each variable $v \in V$, and a weight function $\mathbf{w} : V \rightarrow [0, 1]$ with $\sum_{v \in V} \mathbf{w}(v) = 1$. An assignment of \mathcal{I} is a mapping f defined on V with $f(x) \in A_x$ for each variable $x \in V$. Each instance \mathcal{I} of an assignment problem has associated with it a notion of a *satisfying assignment*. For two assignments f and g for \mathcal{I} , we define their distance as $\text{dist}(f, g) := \sum_{x \in V: f(x) \neq g(x)} \mathbf{w}(x)$. We define $\text{dist}_{\mathcal{I}}(f) = \min_g \text{dist}(f, g)$, where g is over all satisfying assignments of \mathcal{I} . Then, for $\epsilon \in [0, 1]$, we say that an assignment f for \mathcal{I} is ϵ -far from satisfying \mathcal{I} if $\text{dist}_{\mathcal{I}}(f) > \epsilon$. In the *assignment testing problem* corresponding to an assignment problem, we are given an instance \mathcal{I} of the assignment problem and a query access to an assignment f for \mathcal{I} , that is, we can obtain the value of $f(x)$ by querying $x \in V$. Then, we say that an algorithm is a *tester* for the assignment problem if it accepts with probability at least $2/3$ when f is a satisfying assignment of \mathcal{I} , and rejects with probability at least $2/3$ when f is ϵ -far from satisfying \mathcal{I} . The *query complexity* of a tester is the number of queries to f .

We can naturally view $\text{CSP}(\mathbf{A})$ and $\exists\text{CSP}(\mathbf{A})$ as assignment problems: for each instance on a set of (free) variables V , the associated assignments are the mappings from V to A , and the notion of satisfying assignments is as described above. Note that an input to the assignment testing problem corresponding to $\text{CSP}(\mathbf{A})$ or to $\exists\text{CSP}(\mathbf{A})$ is a tuple $(\mathcal{I}, \epsilon, f)$, where \mathcal{I} is an instance of $\text{CSP}(\mathbf{A})$ or $\exists\text{CSP}(\mathbf{A})$, respectively, ϵ is an error parameter, and f is an assignment to \mathcal{I} . In order to distinguish \mathcal{I} from the tuple $(\mathcal{I}, \epsilon, f)$, we always call the former *instance* and the latter *input*.

2.1.1 Gap-preserving local reductions

We will frequently use the following reduction when constructing algorithms as well as showing lower bounds.

Definition 2.1 (Gap-preserving local reduction). *Given assignment problems \mathcal{P} and \mathcal{P}' , there is a (randomized) gap-preserving local reduction from \mathcal{P} to \mathcal{P}' if there exist a function $t(n)$ and constants c_1, c_2 satisfying the following: given a \mathcal{P} -instance \mathcal{I} of with variable set V and an assignment f for \mathcal{I} , there exist a \mathcal{P}' -instance \mathcal{I}' with variable set V' and an assignment f' for \mathcal{I}' such that the following hold:*

1. $|V'| \leq t(|V|)$.
2. If f is a satisfying assignment of \mathcal{I} , then f' is a satisfying assignment of \mathcal{I}' .
3. For any $\epsilon \in (0, 1)$, if $\text{dist}_{\mathcal{I}}(f) \geq \epsilon$, then $\Pr[\text{dist}_{\mathcal{I}'}(f') \geq c_1\epsilon] \geq 9/10$ holds, where the probability is over internal randomness.
4. Any query to f' can be answered by making at most c_2 queries to f .

A *linear reduction* is defined to be a gap-preserving local reduction for which the function $t(n) = O(n)$, $c_1 = O(1)$, and $c_2 = O(1)$.

Lemma 2.2 ([33]). *Let \mathcal{P} and \mathcal{P}' be assignment problems. Suppose that there exists an ϵ -tester for \mathcal{P}' with query complexity $q(n, \epsilon)$ for any $\epsilon \in (0, 1)$, where n is the number of variables in the given instance of \mathcal{P}' , and that there exists a gap-preserving local reduction from \mathcal{P} to \mathcal{P}' with a function t and $c_1 = c_2 = O(1)$. Then, there exists an ϵ -tester for \mathcal{P} with query complexity $O(q(t(n), O(\epsilon)))$ for any $\epsilon > 0$, where n is the number of variables in the given instance of \mathcal{P} . In particular, linear reductions preserve constant-query and sublinear-query testability.*

As another application of gap-preserving local reductions, the following fact is known.

Lemma 2.3 (Lemma 6.4 and 6.5 of [33]). *Let \mathbf{A}, \mathbf{A}' be relational structures. If the relations of \mathbf{A} are preserved by the operations of some finite algebra in $\mathcal{V}(\text{Alg}(\mathbf{A}'))$, then $\text{CSP}(\mathbf{A})$ is constant-query testable if $\text{CSP}(\mathbf{A}')$ is constant-query testable.*

In the proof of Lemma 2.3, the only obstacle that prevents linear reductions is that the number of variables blows up by introducing new variables for each constraint. However, we can get rid of this obstacle by replacing them with existentially quantified variables and we get the following.

Lemma 2.4. *Let \mathbf{A}, \mathbf{A}' be relational structures. If the relations of \mathbf{A} are preserved by the operations of some finite algebra in $\mathcal{V}(\text{Alg}(\mathbf{A}'))$, then there exists a linear reduction from $\exists\text{CSP}(\mathbf{A})$ to $\exists\text{CSP}(\mathbf{A}')$. In particular, $\exists\text{CSP}(\mathbf{A})$ is constant-query (resp., sublinear-query) testable if $\exists\text{CSP}(\mathbf{A}')$ is constant-query (resp., sublinear-query) testable.*

2.1.2 $(k+1)$ -ary near unanimity polymorphisms

Let $\mathcal{I} = (V, V^\exists, \mathcal{C}, \mathbf{w})$ be an instance of $\exists\text{CSP}(\mathbf{A})$. A *partial solution* of \mathcal{I} on a set of variables $W \subseteq V \cup V^\exists$ is a mapping $\psi : W \rightarrow A$ that satisfies every constraint $\langle W \cap s, \text{pr}_{W \cap s} R \rangle$ where $\langle s, R \rangle \in \mathcal{C}$ and $\text{pr}_{W \cap s} R$ is the projection of R to $W \cap s$. Here $W \cap s$ denotes the subtuple of s consisting of those entries of s that belong to W , and we consider the coordinate positions of R indexed by variables from s . Instance \mathcal{I} is said to be *k-consistent* if for any k -element set $W \subseteq V \cup V^\exists$ and any $v \in (V \cup V^\exists) \setminus W$ any partial solution on W can be extended to a partial solution on $W \cup \{v\}$. It is well known that, for any constant k , any CSP instance can be transformed to a k -consistent instance in polynomial time without changing the set of satisfying assignments. See [22] for more details.

Let $\mathbf{A} = (A; \Gamma)$ be a relational structure with a $(k+1)$ -ary near unanimity polymorphism, and let $\mathcal{I} = (V, V^\exists, \mathcal{C}, \mathbf{w})$ be a k -consistent instance of $\exists\text{CSP}(\mathbf{A})$. Because of the existence of a $(k+1)$ -ary near unanimity polymorphism, we can assume that every constraint is k -ary. Hence, we can write $\mathcal{I} = (V, V^\exists, \{R_S\}_{(V \cup V^\exists)^k}, \mathbf{w})$. Further, we can say that, for any set $S \subseteq V \cup V^\exists$ of size k , any partial assignment $f : S \rightarrow A$ with $f|_S \in R_S$ can be extended to a satisfying assignment for the whole instance [16]. This property is called the *k-Helly property*. We call a subset of variables $S \subseteq V$ of size k *violated* with respect to an assignment $f : V \rightarrow A$ if $f|_S \notin R_S$.

As an application of gap-preserving local reductions, we observe that if a relational structure \mathbf{A} has a $(k+1)$ -ary near unanimity polymorphism for some $k \geq 2$, then testing $\exists\text{CSP}(\mathbf{A})$ can be reduced to $\text{CSP}(\mathbf{A})$.

Lemma 2.5. *Let \mathbf{A} be a relational structure with a $(k+1)$ -ary near unanimity polymorphism for some $k \geq 2$. Then, there is a linear reduction from $\exists\text{CSP}(\mathbf{A})$ to $\text{CSP}(\mathbf{A})$.*

Proof. Let $\mathcal{I} = (V, V^\exists, \{R_S\}_{S \in (V \cup V^\exists)^k}, \mathbf{w})$ be a k -consistent instance of $\exists\text{CSP}(\mathbf{A})$. Then, we consider the instance $\mathcal{I}' = (V, \{R_S\}_{S \subseteq V^k}, \mathbf{w})$ and the assignment $f' = f$.

If f satisfies \mathcal{I} , then f' also satisfies \mathcal{I}' because the constraints of \mathcal{I}' are also constraints of \mathcal{I} .

Suppose that f' is ϵ -close to satisfying \mathcal{I}' and let g' be a satisfying assignment of \mathcal{I}' with $\text{dist}_{\mathcal{I}'}(f', g') \leq \epsilon$. Then, we define $g = g'$. Note that g satisfies \mathcal{I} because there is no violated constraint caused by g , and from the k -Helly property, we can always extend it to a satisfying assignment for the whole instance. Hence, f is ϵ -close to g .

To summarize, this reduction is a gap-preserving local reduction with $t(n) = n$, $c_1 = 1$, and $c_2 = 1$. \square

3 Constant-Query Testability

In this section, assume that $\mathbf{A} = (A; \Gamma)$ is a structure that has a majority polymorphism $m(x, y, z)$ and a Maltsev polymorphism $p(x, y, z)$. It is known, [12], that this is equivalent to the variety \mathcal{A} generated by the algebra $\text{Alg}(\mathbf{A})$ being *congruence distributive* and *congruence permutable* and also to \mathbf{A} having a $(k+1)$ -ary near unanimity polymorphism for some $k \geq 2$ and a Maltsev polymorphism. This means that for each algebra $\mathbb{B} \in \mathcal{A}$, the lattice of congruences of \mathbb{B} satisfies the distributive law and that for each pair of congruences α and β of \mathbb{B} , the relations $\alpha \circ \beta$ and $\beta \circ \alpha$ are equal. Such varieties are also said to be *arithmetic*.

An important feature of \mathcal{A} (and in fact of any congruence distributive variety generated by a finite algebra) is that every subdirectly irreducible member of \mathcal{A} has size bounded by $|A|$ ([12]). We

will make use of the fact that an algebra is *subdirectly irreducible* if and only if the intersection of all of its non-trivial congruences is non-trivial. This is equivalent to the algebra having a smallest non-trivial congruence. In this section, we will show that $\exists\text{CSP}(\mathbf{A})$ is constant-query testable. Some of the ideas found in this section were inspired by the paper [9].

We first note that, since \mathbf{A} has a majority operation, that is, a 3-ary near unanimity operation, as a polymorphism, it suffices to consider $\text{CSP}(\mathbf{A})$ by Lemma 2.5.

For our analysis, it is useful to introduce $\text{CSP}(\mathcal{V})$ for a variety \mathcal{V} . An instance of $\text{CSP}(\mathcal{V})$ is of the form $(V, \{A_x\}_{x \in V}, \mathcal{C}, \mathbf{w})$. Each A_x is the domain of an algebra, denoted by \mathbb{A}_x , in \mathcal{V} , and each constraint in \mathcal{C} is of the form $\langle (x_1, \dots, x_k), R \rangle$, where R is the domain of a subalgebra \mathbb{R} of $\mathbb{A}_{x_1} \times \dots \times \mathbb{A}_{x_k}$. In particular, R is also the domain of an algebra in \mathcal{V} . The definitions of 2-consistency and an assignment testing problem naturally carry over to instances of $\text{CSP}(\mathcal{V})$.

Let $\mathcal{I} = (V, \{A_x\}_{x \in V}, \mathcal{C}, \mathbf{w})$ be an instance of $\text{CSP}(\mathcal{A})$. Since \mathcal{A} is arithmetic, we can assume that each constraint in \mathcal{C} is binary [1]. Hence, we also write

$$\mathcal{I} = (V, \{A_x\}_{x \in V}, \{R_{xy}\}_{(x,y) \in V^2}, \mathbf{w})$$

or simply $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$. Moreover, we can assume that \mathcal{I} is 2-consistent because the set of satisfying assignments does not change after making \mathcal{I} 2-consistent. For $x \in V$, R_{xx} is the equality relation 0_{A_x} on the set A_x , and for distinct variables $x \neq y \in V$, R_{xy} denotes the (unique) binary constraint relation from A_x to A_y . We always have $R_{yx} = R_{xy}^{-1} = \{(b, a) \mid (a, b) \in R_{xy}\}$ for any $x, y \in V$. We note that by 2-consistency, it follows that for distinct variables x and y , the relation $R_{x,y}$ is subdirect in $A_x \times A_y$. Throughout the remainder of this section, we will assume that any instance of $\text{CSP}(\mathcal{A})$ considered will be 2-consistent and has only binary constraints.

Since \mathcal{A} is assumed to be congruence permutable (then for any $x \neq y \in V$, the binary relation R_{xy} is *rectangular*, that is, $(a, c), (a, d), (b, d) \in R_{xy}$ implies $(b, c) \in R_{xy}$). As noted in Lemma 2.10 of [9], this is equivalent to R_{xy} being a *thick mapping*. This means that there are congruences θ_{xy} of \mathbb{A}_x and θ_{yx} of \mathbb{A}_y such that modulo the congruence $\theta_{xy} \times \theta_{yx}$ on \mathbb{R}_{xy} , the relation R_{xy} is the graph of an isomorphism ϕ_{xy} from \mathbb{A}_x/θ_{xy} to \mathbb{A}_y/θ_{yx} and such that for all $a \in A_x$ and $b \in A_y$, $(a, b) \in R_{xy}$ if and only if $\phi_{xy}(a/\theta_{xy}) = b/\theta_{yx}$. In this situation, we say that R_{xy} is a thick mapping with respect to θ_{xy} , θ_{yx} and ϕ_{xy} . For future reference, we note that if for some variables $x \neq y$, the congruence $\theta_{xy} = 0_{A_x}$ then the relation R_{yx} is the graph of a surjective homomorphism from \mathbb{A}_y to \mathbb{A}_x .

3.1 A factoring reduction

Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ be an instance of $\text{CSP}(\mathcal{A})$ and for each $x \in V$ let $\mu_x = \bigwedge_{y \neq x} \theta_{xy}$, a congruence of \mathbb{A}_x . We say that A_x is *prime* if μ_x is the equality congruence 0_{A_x} and *factorable* otherwise. Roughly speaking, if A_x is not prime, then we can factor A_x by μ_x without changing the problem, because no constraint of \mathcal{I} distinguishes values within any μ_x -class. Formally, we define the factoring reduction as in Algorithm 1.

Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$ and let $(\mathcal{I}', \epsilon', f') = \text{FACTOR}(\mathcal{I}, \epsilon, f)$. It is clear that since the instance \mathcal{I} of $\text{CSP}(\mathcal{A})$ is assumed to be 2-consistent then the instance \mathcal{I}' will also be 2-consistent. Furthermore, the sizes of the domains of \mathcal{I}' are no larger than the sizes of the domains of \mathcal{I} . Now we show that the factoring reduction is a linear reduction.

Lemma 3.1. *Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$ and let $(\mathcal{I}', \epsilon', f') = \text{FACTOR}(\mathcal{I}, \epsilon, f)$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

Algorithm 1

```
1: procedure FACTOR( $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w}), \epsilon, f$ )
2:   for  $x \in V$  do
3:      $A_x \leftarrow A_x / \mu_x$ .
4:      $f(x) \leftarrow f(x) / \mu_x$ .
5:   for  $(x, y) \in V \times V$  do
6:      $R_{xy} \leftarrow \{(a/\mu_x, b/\mu_y) \mid (a, b) \in R_{xy}\}$ .
7:   return  $(\mathcal{I}, \epsilon, f)$ .
```

Proof. We show that the factoring reduction is a linear reduction. Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ and $\mathcal{I}' = (V', \{A'_x\}_{x \in V}, \{R'_{xy}\}, \mathbf{w}')$ be the original instance and the reduced instance, respectively.

Note that $|V'| = |V|$ and we can determine the value of $f'(x)$ by querying $f(x)$.

If f satisfies \mathcal{I} , then f' also satisfies \mathcal{I}' . Suppose that f' is ϵ -close to satisfying \mathcal{I}' and let g' be a satisfying assignment of \mathcal{I}' with $\text{dist}_{\mathcal{I}'}(f', g') \leq \epsilon$. Then, we define g to be any assignment for \mathcal{I} such that for $x \in V$, $g(x)$ is taken to be an arbitrary element in the μ_x -class $g'(x)$. Then, g satisfies \mathcal{I} and $\text{dist}_{\mathcal{I}'}(f, g) = \text{dist}_{\mathcal{I}}(f', g') \leq \epsilon$.

To summarize, the factoring reduction is a gap-preserving local reduction with $t(n) = n$, $c_1 = 1$, and $c_2 = 1$. \square

3.2 Reduction to instances with subdirectly irreducible domains

In this section, we provide a reduction that produces instances whose domains are all subdirectly irreducible. Suppose that \mathbb{A} is a subdirect product of two algebras $\mathbb{A}_1, \mathbb{A}_2$ from \mathcal{A} and that \mathbb{B} is a subdirect product of \mathbb{A} and \mathbb{B} for some $\mathbb{B} \in \mathcal{A}$. We can project the relation R onto the factors of \mathbb{A} to obtain two new binary relations from A_1 to B and from A_2 to B , respectively:

$$\begin{aligned} R_1 &= \{(a_1, b) \mid \text{there is some } (a_1, c_2) \in A \text{ with } ((a_1, c_2), b) \in R\}, \\ R_2 &= \{(a_2, b) \mid \text{there is some } (c_1, a_2) \in A \text{ with } ((c_1, a_2), b) \in R\}. \end{aligned}$$

The following shows that the relation R can be recovered from the relations R_1, R_2 , and A (considered as a relation from A_1 to A_2).

Lemma 3.2. *For all $a_1 \in A_1$, $a_2 \in A_2$, and $b \in B$, the following are equivalent:*

- $((a_1, a_2), b) \in R$
- $(a_1, b) \in R_1$, $(a_2, b) \in R_2$ and $(a_1, a_2) \in A$.

Proof. One direction of this claim follows by construction. For the other, suppose that $(a_1, b) \in R_1$, $(a_2, b) \in R_2$ and $(a_1, a_2) \in A$. Then there are elements $c_i \in A_i$, for $i = 1, 2$, with $(a_1, c_2), (c_1, a_2) \in A$, $((a_1, c_2), b), ((c_1, a_2), b) \in R$. Since R is subdirect in $A \times B$ and $(a_1, a_2) \in A$ then there is some $d \in B$ with $((a_1, a_2), d) \in R$. Applying the majority term of \mathcal{A} coordinate-wise to the tuples $((a_1, c_2), b)$, $((c_1, a_2), b)$, and $((a_1, a_2), d)$ from R we produce the tuple $((a_1, a_2), b) \in R$, as required. \square

Lemma 3.2 allows us to split a domain of an instance of $\text{CSP}(\mathcal{A})$ into subdirectly irreducible domains. Formally, we define the splitting reduction as in Algorithm 2.

Algorithm 2

```

1: procedure SPLIT( $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w}), \epsilon, f$ )
2:   while there exists  $x \in V$  such that  $\mathbb{A}_x$  is not subdirectly irreducible or trivial do
3:     Replace  $\mathbb{A}_x$  in  $\mathcal{I}$  with an isomorphic non-trivial subdirect product of  $\mathbb{A}_x^1 \times \mathbb{A}_x^2$  for some
       quotients  $\mathbb{A}_x^1, \mathbb{A}_x^2$  of  $\mathbb{A}_x$  such that  $\mathbb{A}_x^1$  is subdirectly irreducible.
4:      $V \leftarrow (V \setminus \{x\}) \cup \{x_1, x_2\}$ , where  $x_1$  and  $x_2$  are newly introduced variables.
5:     Remove the domain  $A_x$  and add the domains  $A_{x_1}^1$  and  $A_{x_2}^2$  over the variables  $x_1$  and  $x_2$ 
       respectively.
6:      $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\langle (x, x), R_{xx} \rangle, \langle (x, y), R_{xy} \rangle, \langle (y, x), R_{yx} \rangle\}_{y \in V \setminus \{x\}}$ .
7:      $\mathcal{C} \leftarrow \mathcal{C} \cup \{\langle (x_1, x_1), 0_{A_{x_1}^1} \rangle, \langle (x_2, x_2), 0_{A_{x_2}^2} \rangle, \langle (x_1, x_2), A_x \rangle, \langle (x_2, x_1), A_x^{-1} \rangle\}$ .
8:      $\mathcal{C} \leftarrow \mathcal{C} \cup \{\langle (x_1, y), (R_{xy})_1 \rangle, \langle (x_2, y), (R_{xy})_2 \rangle, \langle (y, x_1), (R_{xy})_1^{-1} \rangle, \langle (y, x_2), (R_{xy})_2^{-1} \rangle\}_{y \in V \setminus \{x\}}$ .
9:     Remove  $x$  from the domain of  $\mathbf{w}$  and add  $x_1$  and  $x_2$ .
10:    Set  $\mathbf{w}(x_1) = \mathbf{w}(x)/2$  and  $\mathbf{w}(x_2) = \mathbf{w}(x)/2$ .
11:    Remove  $x$  from the domain of  $f$  and add  $x_1$  and  $x_2$ .
12:    Set  $f(x_1) \in A_{x_1}^1$  and  $f(x_2) \in A_{x_2}^2$  so that  $(f(x_1), f(x_2)) = f(x)$ .
13:  return  $(\mathcal{I}, \epsilon/2^{|A|}, f)$ .
```

Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$ and let $(\mathcal{I}', \epsilon', f') = \text{SPLIT}(\mathcal{I}, \epsilon, f)$. It is clear that, since \mathcal{I} is assumed to be a 2-consistent instance of $\text{CSP}(\mathcal{A})$ then the splitting reduction constructs another 2-consistent instance \mathcal{I}' of $\text{CSP}(\mathcal{A})$ whose domains are all subdirectly irreducible and so have size bounded by $|A|$ (and are no bigger than the domains of \mathcal{I}). The next lemma shows that if a domain of an instance \mathcal{I} is prime, then after splitting it, the resulting subdirect factors will also be prime.

Lemma 3.3. *Let \mathcal{I}' be the instance of $\text{CSP}(\mathcal{A})$ obtained by splitting a domain \mathbb{A}_x of another instance \mathcal{I} into two subdirect factors \mathbb{A}_{x_1} and \mathbb{A}_{x_2} as in the SPLIT procedure. If the domain \mathbb{A}_x is prime in \mathcal{I} then the domains \mathbb{A}_{x_1} and \mathbb{A}_{x_2} are prime in \mathcal{I}' .*

Proof. Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ be given and suppose that the domain \mathbb{A}_x is a subdirect product of the algebras \mathbb{A}_{x_1} and \mathbb{A}_{x_2} . To produce \mathcal{I}' from \mathcal{I} by splitting \mathbb{A}_x , we replace the variable x and the domain A_x with the variables x_1 and x_2 and the corresponding domains A_{x_1} and A_{x_2} . For each $y \in V$ with $x \neq y$, we replace the constraint $\langle (x, y), R_{xy} \rangle$ with the constraints $\langle (x_1, y), (R_{xy})_1 \rangle$ and $\langle (x_2, y), (R_{xy})_2 \rangle$ and add the constraint $\langle (x_1, x_2), A_x \rangle$.

If the domain \mathbb{A}_x is prime in \mathcal{I} then there is $k \geq 1$ and variables $y_i \in V \setminus \{x\}$, for $1 \leq i \leq k$, such that $\bigwedge_{1 \leq i \leq k} \theta_{xy_i} = 0_{A_x}$. To show that \mathbb{A}_{x_1} is prime in \mathcal{I}' it will suffice to show that

$$\left(\bigwedge_{1 \leq i \leq k} \theta_{x_1 y_i} \right) \wedge \theta_{x_1 x_2} = 0_{A_{x_1}}.$$

To establish this, suppose that (a_1, a'_1) belongs to the left hand side of this equality. We will show that $a_1 = a'_1$. We have that $(a_1, a'_1) \in \theta_{x_1 y_i}$ for $1 \leq i \leq k$ and $(a_1, a'_1) \in \theta_{x_1 x_2}$. From the latter membership it follows that there is some $c \in A_{x_2}$ such that $(a_1, c), (a'_1, c) \in A_x$. From $(a_1, a'_1) \in \theta_{x_1 y_i}$ it follows that there is some $u \in A_{y_i}$ with $(a_1, u), (a'_1, u) \in (R_{xy_i})_1$. We can conclude that there are $d, d' \in A_{y_i}$ with $((a_1, d), u), ((a'_1, d'), u) \in R_{xy_i}$. We then have that $((a_1, d), (a'_1, d')) \in \theta_{xy_i}$. We can now apply the majority term of \mathcal{A} coordinate-wise to the following three pairs of members of θ_{xy_i} to establish that $((a_1, c), (a'_1, c)) \in \theta_{xy_i}$: $((a_1, d), (a'_1, d'))$, $((a_1, c), (a_1, c))$, and $((a'_1, c), (a'_1, c))$. We've

shown that (a_1, c) and (a'_1, c) are θ_{xy_i} -related for all $i \leq k$ and so we have that $(a_1, c) = (a'_1, c)$, which implies that $a_1 = a'_1$, as required. Thus \mathbb{A}_{x_1} is prime in \mathcal{I}' and by symmetry, \mathbb{A}_{x_2} is also prime. \square

Now we show that the splitting reduction is a gap-preserving local reduction.

Lemma 3.4. *Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$ and let $(\mathcal{I}', \epsilon', f') = \text{SPLIT}(\mathcal{I}, \epsilon, f)$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

Proof. We show that the splitting reduction is a linear reduction.

Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ and $\mathcal{I}' = (V', \{A'_x\}, \{R'_{xy}\}, \mathbf{w}')$ be the original instance and the reduced instance, respectively.

In the reduction, every variable x of V is ultimately split into variables x_1, \dots, x_{k_x} from V' and the domain \mathbb{A}_x is replaced by subdirectly irreducible domains $\mathbb{A}_x^1, \dots, \mathbb{A}_x^{k_x}$ corresponding to these variables such that \mathbb{A}_x is isomorphic to a subdirect product of these new domains. Since each of the domains has size bounded by $|A|$, then $k_x \leq |A|$ for all $x \in V$ and so after completely splitting \mathbb{A}_x into the k_x factors, we have that $\mathbf{w}(x) \leq 2^{|A|} \mathbf{w}'(x_i)$ for each $i \in [k_x]$. We also have that $\sum_{i \in [k_x]} \mathbf{w}'(x_i) = \mathbf{w}(x)$ for each $x \in V$.

We can determine the value of $f'(x_i)$, where x_i is added when splitting the variable x , we only need to know the value of $f(x)$.

If f satisfies \mathcal{I} , then f' satisfies \mathcal{I}' by Lemma 3.2. Suppose that f' is $\epsilon/(2^{|A|})$ -close to satisfying \mathcal{I}' and let g' be a satisfying assignment for \mathcal{I}' with $\text{dist}(f', g') \leq \epsilon/(2^{|A|})$. Because the tuple $(g'(x_1), \dots, g'(x_{k_x}))$ is in A_x , we can naturally define an assignment g for \mathcal{I} by setting $g(x) = (g'(x_1), \dots, g'(x_{k_x})) \in A_x$. Then g is a satisfying assignment from Lemma 3.2. Moreover,

$$\text{dist}(f, g) = \sum_{x \in V: \exists i \in [k_x], g'(x_i) \neq f'(x_i)} \mathbf{w}(x) \leq \sum_{x \in V} \sum_{i \in [k_x]: g'(x_i) \neq f'(x_i)} 2^{|A|} \mathbf{w}'(x_i) = 2^{|A|} \text{dist}(f', g') \leq \epsilon.$$

To summarize, the splitting reduction is a gap-preserving local reduction with $t(n) = |A|$, $c_1 = 1$, and $c_2 = 2^{|A|}$. \square

3.3 Isomorphism reduction

By applying the factoring reduction and then the splitting reduction to an instance of $\text{CSP}(\mathcal{A})$ we end up with an instance whose domains are either trivial or subdirectly irreducible and prime. For such an instance, we have the following property.

Lemma 3.5. *Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ be an instance of $\text{CSP}(\mathcal{A})$ such that $|V| > 1$ and such that every domain is either trivial or is subdirectly irreducible and prime. Then, for each variable $x \in V$, there is at least one variable $y \neq x$ so that $\theta_{xy} = 0_{A_x}$ and for such variables y , the relation R_{yx} is the graph of a surjective homomorphism from \mathbb{A}_y to \mathbb{A}_x .*

Proof. If $|A_x| = 1$ then the result follows trivially. Otherwise, we have that the congruence $\mu_x = \bigwedge_{y \neq x} \theta_{xy}$ of \mathbb{A}_x is equal to 0_{A_x} , since \mathbb{A}_x is prime. But, since this algebra is subdirectly irreducible, it follows that for some $y \neq x$, $\theta_{xy} = 0_{A_x}$. Since R_{yx} is a thick mapping with $\theta_{xy} = 0_{A_x}$ it follows that R_{yx} is the graph of a surjective homomorphism from \mathbb{A}_y to \mathbb{A}_x . \square

Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ be an instance of $\text{CSP}(\mathcal{A})$ with $|V| > 1$ and with the property that every domain is either trivial or is subdirectly irreducible and prime. Define the relation \sim on V by $x \sim y$ if and only if the relation R_{xy} is the graph of an isomorphism from \mathbb{A}_x to \mathbb{A}_y . Using the 2-consistency of \mathcal{I} , the relation \sim is naturally an equivalence relation on V . The following corollary to Lemma 3.5 establishes that unless all of the domains of \mathcal{I} are trivial, the relation \sim is non-trivial.

Corollary 3.6. *For $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ an instance of $\text{CSP}(\mathcal{A})$ as in Lemma 3.5, if $x \in V$ is such that the domain A_x has maximal size and has at least two elements, then there is some $y \in V$ with $x \neq y$ and $x \sim y$.*

Proof. If A_x has maximal size and has at least two elements, then let $y \in V$ be a variable such that $x \neq y$ and R_{yx} the graph of a surjective homomorphism from \mathbb{A}_y to \mathbb{A}_x . Since A_x has maximal size, it follows that $|A_y| = |A_x|$ and so R_{yx} is the graph of an isomorphism from \mathbb{A}_y to \mathbb{A}_x . \square

For a variable $x \in V$, let $[x] := x / \sim$ denote the \sim -class of V that x belongs to. Let $S \subseteq V$ be an arbitrary complete system of representatives of this equivalence relation and for any \sim -class u , let $s(u) \in V$ be the unique element $x \in S$ such that $x \in u$. In particular $[s(u)] = u$ holds.

Given an assignment f for \mathcal{I} , we can test the input $(\mathcal{I}, \epsilon, f)$ in two steps. First, we test whether the values of f in the \sim -classes of V are consistent using a consistency algorithm (Algorithm 3) and then we test the input obtained by contracting the \sim -classes using Algorithm 4. Explanations of these two steps are contained in the next two subsections.

3.3.1 Testing \sim -consistency

We say that the input $(\mathcal{I}, \epsilon, f)$ is \sim -consistent if, for each $x, y \in V$ with $x \sim y$, $(f(x), f(y)) \in R_{xy}$.

For a \sim -class $u \subseteq V$ and $b \in A_{s(u)}$, we define

$$\begin{aligned}\overline{\mathbf{w}}(u, b) &= \sum_{y \in u: f(y) = R_{s(u)y}(b)} \mathbf{w}(y), \\ \overline{\mathbf{w}}(u) &= \sum_{b \in A_{s(u)}} \overline{\mathbf{w}}(u, b), \text{ and} \\ \overline{\mathbf{w}}_{\text{maj}}(u) &= \max_{b \in A_{s(u)}} \overline{\mathbf{w}}(u, b).\end{aligned}$$

Note that $\overline{\mathbf{w}}(u)$ is also equal to $\sum_{x \in u} \mathbf{w}(x)$, the sum of the weights of the variables in u . In addition, we define ϵ_u to be $(\overline{\mathbf{w}}(u) - \overline{\mathbf{w}}_{\text{maj}}(u)) / \overline{\mathbf{w}}(u)$ and observe that $\epsilon_u \leq (|A| - 1) / |A|$ since $|A_{s(u)}| \leq |A|$ and so $\overline{\mathbf{w}}(u)$ is the sum of at most $|A|$ terms, each of which is at most $\overline{\mathbf{w}}_{\text{maj}}(u)$. The quantity ϵ_u represents the fraction of values, by weight, of $f|_u$ that need to be altered in order to establish \sim -consistency of the assignment over the class u . Let f_{maj} be the assignment obtained from f in this way. That is, for $x \in V$, $f_{\text{maj}}(x) = R_{s([x])x} \left(\arg\max_{b \in A_{s([x])}} \overline{\mathbf{w}}([x], b) \right)$.

We need the following simple proposition to analyze our algorithm.

Proposition 3.7. *Let X be a random variable taking values in $[0, 1]$ such that $\mathbf{E}[X] \geq \epsilon$ for some $\epsilon \geq 0$. Then, $\Pr[X \geq \epsilon/2] \geq \epsilon/2$ holds.*

Proof. Let $p = \Pr[X \geq \epsilon/2]$. Then,

$$\epsilon \leq \mathbf{E}[X] \leq 1 \cdot p + \frac{\epsilon}{2}(1 - p) \leq p + \frac{\epsilon}{2}.$$

Hence, $p \geq \epsilon/2$ holds. \square

Algorithm 3

```
1: procedure CONSISTENCY( $\mathcal{I}, \epsilon, f$ )
2:   Sample a set  $U$  of  $\Theta(1/\epsilon)$   $\sim$ -classes of  $\mathcal{I}$ . In each sampling,  $u$  is chosen with probability  $\overline{w}(u)$ .
3:   for each  $u \in U$  do
4:     Sample a set  $S$  of  $\Theta(1/\epsilon)$  variables in  $u$ . In each sampling, a variable  $x \in u$  is chosen with probability  $w(x)/\overline{w}(u)$ .
5:     if there are two variables  $x, y \in S$  with  $f(y) \neq R_{xy}(f(x))$  then
6:       Reject.
7:   Accept.
```

In order to test \sim -consistency, we run Algorithm 3.

Lemma 3.8. *Algorithm 3 tests \sim -consistency with query complexity $O(1/\epsilon^2)$.*

Proof. It is clear that Algorithm 3 accepts if f is \sim -consistent and the query complexity is $O(1/\epsilon^2)$. Suppose that f is ϵ -far from \sim -consistency, which means that $\text{dist}(f, f_{\text{maj}}) \geq \epsilon$. Then, we have $\mathbf{E}[\epsilon_u] = \sum_{u: \sim\text{-class}} (\overline{w}(u))\epsilon_u \geq \epsilon$, where in the calculation of the expectation, a \sim -class u is chosen with probability $\overline{w}(u)$. Note that $\epsilon_u \in [0, 1]$ for every \sim -class u and so we can apply Lemma 3.7, to conclude that we sample a \sim -class u with $\epsilon_u \geq \epsilon/2$ with probability at least $\epsilon/2$. Hence, the probability that U contains a \sim -class u with $\epsilon_u \geq \epsilon/2$ is at least $1 - (1 - \epsilon/2)^{\Theta(1/\epsilon)} \geq 5/6$ by choosing the hidden constant large enough. For a \sim -class u with $\epsilon_u \geq \epsilon/2$, the probability that we find two vertices $x, y \in u$ with $f(y) \neq R_{xy}(f(x))$ in S is at least

$$1 - (1 - \epsilon_u)^{\Theta(1/\epsilon)} - (\epsilon_u)^{\Theta(1/\epsilon)} \geq 1 - (1 - \epsilon/2)^{\Theta(1/\epsilon)} - ((|A| - 1)/|A|)^{\Theta(1/\epsilon)}$$

since $\epsilon_u \geq \epsilon/2$ for this class u and, as noted earlier, $\epsilon_u \leq (|A| - 1)/|A|$ for every class u . By choosing the hidden constant large enough we can ensure that this value is at least $5/6$. By combining these bounds, we obtain two vertices x, y with $f(y) \neq R_{xy}(f(x))$ with probability at least $2/3$. \square

3.3.2 Isomorphism reduction

Using Algorithm 3, we can reject an input $(\mathcal{I}, \epsilon, f)$ if it is far from satisfying \sim -consistency. In this subsection we will consider a reduction from $(\mathcal{I}, \epsilon, f)$ to another input $(\mathcal{I}', \epsilon', f')$ assuming that $(\mathcal{I}, \epsilon, f)$ is close to satisfying \sim -consistency.

Our reduction, as described in Algorithm 4, contracts the variables in each \sim -class to a single variable from that class. It should be clear that since the instance \mathcal{I} of $\text{CSP}(\mathcal{A})$ is assumed to be 2-consistent, the reduction will produce another 2-consistent instance \mathcal{I}' of $\text{CSP}(\mathcal{A})$. As the next lemma shows, unless the domains of \mathcal{I} all have size one, some of the domains of \mathcal{I}' will no longer be prime.

Lemma 3.9. *Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$ for which domains of \mathcal{I} are either trivial or prime and subdirectly irreducible and let $(\mathcal{I}', \epsilon', f') = \text{ISOMORPHISM}(\mathcal{I}, \epsilon, f)$. If some domain of \mathcal{I} has more than one element, then any domain of \mathcal{I}' of maximal size will not be prime, unless \mathcal{I}' has only one variable.*

Algorithm 4

```

1: procedure ISOMORPHISM( $\mathcal{I}, \epsilon, f$ )
2:   for each  $\sim$ -class  $u$  do
3:     Sample a variable  $x \in u$  with probability  $\mathbf{w}(x)/\overline{\mathbf{w}}(u)$ , and let  $x_u$  be the sampled variable.
4:      $V' \leftarrow V' \cup \{u\}$ .
5:      $A'_u \leftarrow A_{s(u)}$ .
6:      $\mathbf{w}'(u) \leftarrow \overline{\mathbf{w}}(u)$ .
7:      $f'(u) \leftarrow R_{x_us(u)}(f(x_u))$ .
8:   for each pair  $(u, u')$  of  $\sim$ -classes do
9:      $R'_{uu'} \leftarrow R_{x_u x_{u'}}$ .
10:  return  $((V', \{A'_x\}, \{R'_{xy}\}, \mathbf{w}'), \epsilon/2, f')$ .

```

Proof. Suppose that \mathcal{I}' has more than one variable. This is equivalent to there being more than one \sim -class for \mathcal{I} . Let x be a variable of \mathcal{I}' with $|A_x|$ of maximal size and let y be any other variable of \mathcal{I}' . Note that according to the construction of \mathcal{I}' from \mathcal{I} , both x and y are also variables of \mathcal{I} with $x \not\sim y$. Furthermore, $|A_x|$ has maximal size amongst all of the domains of \mathcal{I} and so the relation R_{yx} cannot be the graph of a surjective homomorphism from \mathbb{A}_y to \mathbb{A}_x . If it were, then it would be the graph of an isomorphism, contradicting that $x \not\sim y$. Thus the congruence $\theta_{xy} \neq 0_{A_x}$. Since \mathbb{A}_x is subdirectly irreducible it follows that $\mu_x = \bigwedge_{y \neq x} \theta_{xy}$ is also not equal to 0_{A_x} and so A_x is not prime in \mathcal{I}' . \square

Lemma 3.10. *Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$ and suppose that f is $\epsilon/20$ -close to satisfying \mathcal{I} . Let $(\mathcal{I}', \epsilon', f') = \text{ISOMORPHISM}(\mathcal{I}, f)$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

Proof. We show that the reduction in Algorithm 4 is a linear reduction. Let $\mathcal{I} = (V, \{A_x\}, \{R_{xy}\}, \mathbf{w})$ and $\mathcal{I}' = (V', \{A'_x\}, \{R'_{xy}\}, \mathbf{w}')$ be the original instance and the reduced instance, respectively.

Note that $|V'| \leq |V|$ and we can determine the value of $f'(u)$ by querying $f(x_u)$. Also, if f satisfies \mathcal{I} , then it is clear that f' satisfies \mathcal{I}' .

We want to show that, if f is far from satisfying \mathcal{I} , then f' is also far from satisfying \mathcal{I}' with high probability. To this end, we first show that the following quantity is small with high probability:

$$\text{dist}(f, f') := \sum_{u: \sim\text{-class}} \sum_{\substack{x \in u: \\ f'(u) \neq R_{xs(u)}(f(x))}} \mathbf{w}(x).$$

For a \sim -class u , we define

$$\text{dist}_u(f, f') := \sum_{\substack{x \in u: \\ f'(u) \neq R_{xs(u)}(f(x))}} \frac{\mathbf{w}(x)}{\overline{\mathbf{w}}(u)}.$$

Note that we have $\text{dist}(f, f') = \sum_{u: \sim\text{-class}} \overline{\mathbf{w}}(u) \text{dist}_u(f, f')$.

Algorithm 5

```

1: procedure ISOMORPHISM'( $\mathcal{I}, \epsilon, f$ )
2:   if CONSISTENCY( $\mathcal{I}, \epsilon/20, f$ ) rejects then
3:     Reject.
4:   else
5:     return ISOMORPHISM( $\mathcal{I}, \epsilon, f$ )

```

Then for any \sim -class u ,

$$\begin{aligned}
\mathbf{E}_{x_u}[\text{dist}_u(f, f')] &= \sum_{b \in A_{s(u)}} \frac{\overline{w}(u, b)}{\overline{w}(u)} \left(1 - \frac{\overline{w}(u, b)}{\overline{w}(u)}\right) \\
&\leq \frac{\overline{w}_{\text{maj}}(u)}{\overline{w}(u)} \left(1 - \frac{\overline{w}_{\text{maj}}(u)}{\overline{w}(u)}\right) + \left(1 - \frac{\overline{w}_{\text{maj}}(u)}{\overline{w}(u)}\right) \cdot 1 \\
&\leq 2 \left(1 - \frac{\overline{w}_{\text{maj}}(u)}{\overline{w}(u)}\right) = 2\epsilon_u.
\end{aligned}$$

Thus, $\mathbf{E}_{\{x_u\}_{u: \sim\text{-class}}}[\text{dist}(f, f')]$ is equal to

$$\mathbf{E}_{\{x_u\}} \left[\sum_{u: \sim\text{-class}} \overline{w}(u) \text{dist}_u(f, f') \right] \leq \sum_{u: \sim\text{-class}} 2\overline{w}(u)\epsilon_u < \frac{\epsilon}{10}.$$

Also, for any \sim -class u , $\mathbf{Var}_{x_u}[\text{dist}_u(f, f')]$ is at most

$$\mathbf{E}_{x_u}[\text{dist}_u(f, f')^2] \leq \mathbf{E}[\text{dist}_u(f, f')] \leq 2\epsilon_u.$$

Here we have used the fact that $0 \leq \text{dist}_u(f, f') \leq 1$.

Since random variables $\{\text{dist}_u(f, f')\}_{u: \sim\text{-class}}$ are independent, we have

$$\mathbf{Var}_{\{x_u\}}[\text{dist}(f, f')] = \mathbf{Var}_{\{x_u\}} \left[\sum_{u: \sim\text{-class}} \overline{w}(u) \text{dist}_u(f, f') \right] \leq \sum_{u: \sim\text{-class}} \overline{w}(u)^2 \cdot 2\epsilon_u \leq \sum_{u: \sim\text{-class}} 2\overline{w}(u)\epsilon_u \leq \frac{\epsilon}{10}.$$

From Chebyshev's inequality, we have $\Pr_{\{x_u\}}[\text{dist}(f, f') \geq \epsilon/2] \leq \Pr_{\{x_u\}}[|\text{dist}(f, f') - \epsilon/10| \geq 4 \cdot \epsilon/10] \leq 1/16$.

Let g' be a satisfying assignment for \mathcal{I}' closest to f' . We define an assignment g for \mathcal{I} as $g(x) = R_{s([x])x}g'([x])$. It is clear that g is a satisfying assignment. Since we have $\text{dist}(f, f') + \text{dist}(f', g') \geq \text{dist}(f, g) \geq \epsilon$, it follows that $\Pr[\text{dist}(f', g') \geq \epsilon/2] \geq 15/16$.

To summarize, the isomorphism reduction is a gap-preserving local reduction with $t(n) \leq n$, $c_1 = 1$, and $c_2 = 2$. \square

Finally, we combine Algorithm 3 and Algorithm 4. to produce Algorithm 5 and make use of it in the following.

Lemma 3.11. *Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$ and suppose that $\text{ISOMORPHISM}(\mathcal{I}, f)$ returned another instance $(\mathcal{I}', \epsilon', f')$. If $(\mathcal{I}', \epsilon', f')$ is testable with $q(\epsilon')$ queries, then $(\mathcal{I}, \epsilon, f)$ is testable with $q(O(\epsilon))$ queries.*

Proof. Consider Algorithm 5. If f satisfies \mathcal{I} , then the \sim -consistency test always accepts, and hence we always accept with probability $2/3$ from Lemma 3.10. Suppose that f is ϵ -far from satisfying \mathcal{I} . If f is $\epsilon/20$ -far from satisfying \sim -consistency, then the \sim -consistency test rejects with probability at least $2/3$. If f is $\epsilon/20$ -close to satisfying \sim -consistency, then we reject with probability at least $2/3$ by Lemma 3.10. \square

3.4 Putting things together

Combining the reductions introduced so far we can design a shrinking reduction, which shrinks the maximum size of the domains of an instance of $\text{CSP}(\mathcal{A})$.

Algorithm 6

```

1: procedure SHRINK( $\mathcal{I}, \epsilon, f$ )
2:    $(\mathcal{I}, \epsilon, f) \leftarrow \text{FACTOR}(\mathcal{I}, \epsilon, f)$ .
3:    $(\mathcal{I}, \epsilon, f) \leftarrow \text{SPLIT}(\mathcal{I}, \epsilon, f)$ .
4:   if ISOMORPHISM'( $\mathcal{I}, \epsilon, f$ ) rejects then
5:     Reject.
6:   else
7:      $(\mathcal{I}, \epsilon, f) \leftarrow$  the input returned by ISOMORPHISM'.
8:    $(\mathcal{I}, \epsilon, f) \leftarrow \text{FACTOR}(\mathcal{I}, \epsilon, f)$ .
9:   return  $(\mathcal{I}, \epsilon, f)$ .
```

Lemma 3.12. *Let $(\mathcal{I}, \epsilon, f)$ be an input of $\text{CSP}(\mathcal{A})$, and suppose that $\text{SHRINK}(\mathcal{I}, \epsilon, f)$ returned another instance $(\mathcal{I}', \epsilon', f')$. If we can test $(\mathcal{I}', \epsilon', f')$ with $q(\epsilon')$ queries, then we can test $(\mathcal{I}, \epsilon, f)$ with $q(O(\epsilon))$ queries. Moreover, the reduction reduces the maximum size of a domain of the given input, if this maximum is greater than one and the reduced instance has more than one variable.*

Proof. We note that at each step of the algorithm, the domains of the instances that are produced are no larger than the domains of the original instance. Furthermore, if any of the domains of the original instance has size greater than one, then it follows from Lemma 3.9 that the maximal size of the domains of the output instance will be smaller than that of the original instance, as long as the output instance has more than one variable. \square

Theorem 3.13. *Let \mathbf{A} be a structure that has majority and Maltsev polymorphisms. Then, $\text{CSP}(\mathbf{A})$ and $\exists\text{CSP}(\mathbf{A})$ are constant-query testable with one-sided error.*

Proof. By applying the shrinking reduction at most $|A|$ times, we get an instance for which every variable has a domain of size one or which has only one variable. In either case, the testing becomes trivial. \square

4 Non Constant-Query Testability

In this section we consider structures \mathbf{A} that do not have a majority polymorphism or do not have a Maltsev polymorphism. As noted in the previous section, this is the same as the variety $\mathcal{V}(\text{Alg}(\mathbf{A}))$ failing to be arithmetic. For such structures we will show that $\text{CSP}(\mathbf{A})$ is not constant-query testable.

From [19] we know that for a structure \mathbf{A} , having both majority and Maltsev polymorphisms is equivalent to $\mathcal{V}(\text{Alg}(\mathbf{A}))$ being congruence meet semidistributive and congruence permutable.

First suppose that $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is not congruence meet semidistributive. We observe that $\text{CSP}(\mathbf{A}')$ will be sublinear-query testable if $\text{CSP}(\mathbf{A})$ is, where \mathbf{A}' is obtained from \mathbf{A} by adding all the unary constant relations (see Lemma 5 of [7]). Although the original proof of the Lemma only considers the unweighted Boolean case, it is straightforward to generalize it to the weighted finite domain case, and we do not repeat it here. By adding all of the unary constant relations to \mathbf{A} to produce \mathbf{A}' it follows that the variety $\mathcal{V}(\text{Alg}(\mathbf{A}'))$ is idempotent and will also not be congruence meet semidistributive. For such a structure, it is known that testing $\text{CSP}(\mathbf{A}')$ requires a linear number of queries [33], and hence testing $\text{CSP}(\mathbf{A})$ will also require a linear number of queries.

From the argument above, in order to complete the proof of Theorem 4.4 it suffices to show that $\text{CSP}(\mathbf{A})$ is not constant-query testable when \mathbf{A} does not have a Maltsev polymorphism (or equivalently when $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is not congruence permutable). We use the following fact.

Lemma 4.1. *Let \mathbf{A} be a relational structure that does not have a Maltsev polymorphism. Then, there is some finite algebra \mathbb{B} in $\mathcal{V}(\text{Alg}(\mathbf{A}))$ and some subuniverse γ of \mathbb{B}^2 such that there are elements 0 and $1 \in B$ with $(0, 0), (0, 1), (1, 1) \in \gamma$ and $(1, 0) \notin \gamma$.*

Proof. Since \mathbf{A} does not have a Maltsev polymorphism, then $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is not congruence permutable and so there is some finite algebra $\mathbb{B} \in \mathcal{V}(\text{Alg}(\mathbf{A}))$ having congruences α and β such that $\alpha \circ \beta \neq \beta \circ \alpha$. We may assume that $\alpha \circ \beta \not\subseteq \beta \circ \alpha$ and so there will be elements $0, 1 \in B$ with $(0, 1) \in \alpha \circ \beta$ but $(1, 0) \notin \alpha \circ \beta$. Since $\alpha \circ \beta$ is a reflexive relation, then setting $\gamma = \alpha \circ \beta$ works. \square

We now establish a super-constant lower bound for $\text{CSP}((B; \gamma))$ for B and γ as in Lemma 4.1. Although the argument is similar to a super-constant lower bound for monotonicity testing given in [17], we present it here for completeness.

Let $G = (V; E)$ be an undirected graph and let $M \subseteq E$ be a matching in G , i.e., no two edges in M have a vertex in common. Let $V(M)$ be the set of the endpoints of edges in M . A matching M is called *induced* if the subgraph induced by M contains only the edges of M . A bipartite graph $G = (X, Y; E)$ is called (s, t) -Ruzsá-Szemerédi if its edge set can be partitioned into at least s induced matchings M_1, \dots, M_s , each of size at least t .

Lemma 4.2 (Theorem 16 of [17]). *There exist an $(n^{\Omega(1/\log \log n)}, n/3 - o(n))$ -Ruzsá-Szemerédi graphs $G = (X, Y; E)$ with $|X| = |Y| = n$.*

Theorem 4.3. *Let $\mathbf{B} = (B; \gamma)$ where γ is a binary relation such that for some $0, 1 \in B$, $(0, 1), (0, 0)$, and $(1, 1) \in \gamma$ but $(1, 0) \notin \gamma$. Then, $\text{CSP}(\mathbf{B})$ is not constant-query testable.*

Proof. If $\text{CSP}(\mathbf{B})$ is testable with q queries, then $\text{CSP}(\mathbf{B})$ is non-adaptively testable with $|A|^q$ queries. Hence, in order to show that $\text{CSP}(\mathbf{B})$ is not constant-query testable, it suffices to show that $\text{CSP}(\mathbf{B})$ is not constant-query testable non-adaptively.

Let $G = (X, Y; E)$ be an $(s, n/3 - o(n))$ -Ruzsá-Szemerédi graph provided as in Lemma 4.2, where $s = n^{\Omega(1/\log \log n)}$. Then, we construct an instance $\mathcal{I} = (V, \mathcal{C}, \mathbf{w})$ of $\text{CSP}(\mathbf{B})$, where $V = X \cup Y$, $\mathcal{C} = \{\langle (x, y), \gamma \rangle \mid (x, y) \in E\}$, and $\mathbf{w}(x) = 1/|V|$ for all $x \in V$.

We use Yao's principle, which states that to establish a lower bound on the complexity of a randomized test, it is enough to present an input distribution on which any deterministic test with that complexity is likely to fail. Namely, we define distributions D_P, D_N on positive (satisfying) and negative (far from satisfying) assignments, respectively. Our assignment distribution first

chooses D_P or D_N with equal probability and then draws an assignment according to the chosen distribution. We show that every deterministic non-adaptive test with $q = o(\sqrt{s})$ queries has error probability larger than $1/3$ (with respect to the induced probability on assignments).

We now define the distributions D_P and D_N , as well as the auxiliary distribution \tilde{D}_N . For D_P and D_N , choose a random $i \in \{1, \dots, s\}$ uniformly. For all variables $x \in X$ and $y \in Y$ outside of matching M_i , set $f(x) = 0$ and $f(y) = 1$. For D_P , uniformly choose $f(x) = f(y) = 0$ or $f(x) = f(y) = 1$ independently for all edges $(x, y) \in M_i$. For \tilde{D}_N , uniformly choose $f(x) = 1 - f(y) = 0$ or $f(x) = 1 - f(y) = 1$ independently for all $(x, y) \in M_i$.

Note that D_P is supported only on positive assignments, but \tilde{D}_N is not supported only on negative assignments. However, for n large enough, with probability more than $8/9$ at least $1/3$ of the constraints on the edges of M_i are violated when the assignment is chosen according to \tilde{D}_N , making the assignment $\Omega(1)$ -far from satisfying \mathcal{I} . Denote the latter event by A and define $\tilde{D}_N|_A$, namely, D_N is \tilde{D}_N conditioned on the event A . Note that for \tilde{D}_N , a constraint is violated only if it belongs to M_i , since the matchings are induced.

Given a deterministic non-adaptive test that makes a set V' of q queries, the probability that one or more of the edges of M_i have both endpoints in V' is at most $q^2/(4s)$ for both D_P and D_N . This is because the matchings are disjoint, and the vertex set V' induces at most $q^2/4$ edges of G . For $q = o(\sqrt{s})$, with probability more than $1 - o(1)$, no edge of M_i has both endpoints in V' . Conditioned on any choice of i for which M_i has no such edge, the distribution of $f|_{V'}$ is identical for both D_N and D_P : every vertex outside of M_i is fixed to 0 if it is in X and to 1 if it is in Y , and the value of every other vertex is uniform and independent over $\{0, 1\}$. Let $C(\phi)$ denote the set of assignments consistent with query answers $\phi : V' \rightarrow \{0, 1\}$. Then, we have $\Pr_{D_P}[C(\phi) \mid \text{no edge in } M_i] = \Pr_{\tilde{D}_P}[C(\phi) \mid \text{no edge in } M_i]$. For every tuple of answers ϕ , the error probability under the above conditioning (with negative assignments chosen under \tilde{D}_N rather than D_N) is $1/2$. As the probability of the condition is at least $1 - o(1)$, the overall error probability without the conditioning is at least $1/2 - o(1)$. Since negative assignments are chosen under D_N , not \tilde{D}_N , the success probability is $(1/2 + o(1)) \cdot (\Pr[A]^{-1}) \leq (1/2 + o(1)) \cdot 9/8 \leq 9/16 + o(1)$. Thus, the error probability is $\geq 7/16 - o(1)$. \square

We can now prove the following theorem.

Theorem 4.4. *If the relational structure \mathbf{A} does not have a majority polymorphism or does not have a Maltsev polymorphism, then $\text{CSP}(\mathbf{A})$ is not constant-query testable.*

Proof. As noted earlier, it suffices to establish hardness under the assumption that $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is not congruence meet semidistributive or that \mathbf{A} does not have a Maltsev polymorphism. The discussion prior to Lemma 4.1 handles the former case, while the combination of that Lemma with Theorem 4.3 and Lemma 2.3 handles the latter case. \square

5 Sublinear-Query Testability

Let $\mathbf{A} = (A; \Gamma)$ be a relational structure having, for some $k \geq 2$ a $(k + 1)$ -ary near unanimity polymorphism. In this section, we will show that $\text{CSP}(\mathbf{A})$ and $\exists\text{CSP}(\mathbf{A})$ are sublinear-query testable. From Lemma 2.5, it suffices only to consider $\text{CSP}(\mathbf{A})$. We note that a sublinear-query tester for $\text{CSP}(\mathbf{A})$ is already known for the unweighted case [7], and we will slightly modify their argument to handle weights.

Let $\mathcal{I} = (V, \mathcal{C}, \mathbf{w})$ be an instance of $\text{CSP}(\mathbf{A})$. Since Γ has a $(k+1)$ -ary near-unanimity polymorphism, we can assume that each constraint in \mathcal{C} has arity exactly k [16]. Hence, we can write \mathcal{I} as $(V, \{R_S\}_{S \subseteq V^k}, \mathbf{w})$ or simply $\mathcal{I} = (V, \{R_S\}, \mathbf{w})$. Moreover, we can assume that \mathcal{I} is k -consistent. Recall that a subset of variables $S \subseteq V$ of size k is said to be *violated* with respect to an assignment $f : V \rightarrow A$ if $f|_S \notin R_S$. We have the following fact.

Lemma 5.1. *If an assignment $f : V \rightarrow A$ is ϵ -far from satisfying \mathcal{I} , then there is a family \mathcal{S} of disjoint violated sets (that are contained in V) such that $\sum_{S \in \mathcal{S}} \prod_{x \in S} \mathbf{w}(x) \geq \frac{\epsilon^k}{k2^k n^{k-1}}$.*

Proof. Let V' be the set of variables x such that $\mathbf{w}(x) \geq \epsilon/(2n)$ and let \mathcal{U} be the family of violated sets $S \subseteq V'$ of size k . We say that a subset $H \subseteq V'$ is a *hitting set* of \mathcal{U} if, for any subset $S \in \mathcal{U}$, H and S intersect. We first observe that, for any hitting set $H \subseteq V'$ of \mathcal{U} , the partial assignment $f|_{V' \setminus H}$ is extendable to a satisfying assignment. Indeed, if $f|_{V' \setminus H}$ is not extendable, then there must be a variable set $S \subseteq V' \setminus H$ with $|S| = k$ and $f|_S \notin R_S$ from the Helly property. However, such a set S must be contained in \mathcal{U} , a contradiction.

Since f is ϵ -far and every variable in $V \setminus V'$ has a weight at most $\epsilon/(2n)$, we have $\sum_{x \in H} \mathbf{w}(x) \geq \epsilon - \epsilon/(2n) \cdot n = \epsilon/2$. Then, we can take a family \mathcal{S} of disjoint violated sets such that $\sum_{S \in \mathcal{S}} \prod_{x \in S} \mathbf{w}(x) \geq \epsilon/2$. In particular, this means that

$$\sum_{S \in \mathcal{S}} \prod_{x \in S} \mathbf{w}(x) \geq \left(\frac{\epsilon}{2n}\right)^{k-1} \sum_{S \in \mathcal{S}} \max_{x \in S} \mathbf{w}(x) \geq \frac{1}{k} \left(\frac{\epsilon}{2n}\right)^{k-1} \sum_{S \in \mathcal{S}} \sum_{x \in S} \mathbf{w}(x) \geq \frac{\epsilon}{2k} \left(\frac{\epsilon}{2n}\right)^{k-1} = \frac{\epsilon^k}{k2^k n^{k-1}}. \quad \square$$

Now we establish the main theorem of this section.

Theorem 5.2. *Let \mathbf{A} be a relational structure that has a $(k+1)$ -ary near unanimity polymorphism for some $k \geq 2$. Then, $\text{CSP}(\mathbf{A})$ (and hence $\exists\text{CSP}(\mathbf{A})$) is sublinear-query testable with one-sided error.*

Proof. First, we describe our algorithm. Let $(\mathcal{I} = (V, \mathcal{C}, \mathbf{w}), \epsilon, f)$ be an input of $\text{CSP}(\mathbf{A})$. We query each variable $x \in V$ with probability $q \cdot \mathbf{w}(x)$, where $q = \Theta((\frac{k2^k n^{k-1}}{\epsilon^k})^{1/k}) = \Theta(\frac{2}{\epsilon} k^{1/k} n^{(k-1)/k})$. If we query more than $100q$ times along the way, we immediately stop and accept. Suppose that the number of queries is at most $100q$. Then, we reject if there is some subset $S \subseteq V$ of size k such that $f|_S \notin R_S$, and we accept otherwise. The query complexity is $100q$, which is sublinear in n .

It is easy to see that the algorithm always accepts if f is a satisfying assignment (no matter whether we stopped as we have queried more than $100q$ times).

Now, we see that the algorithm rejects with high probability when the input is ϵ -far. From Markov's inequality, the query complexity is at most $100q$ with probability at least $\frac{99}{100}$.

From Lemma 5.1, there is a family \mathcal{S} of disjoint violated sets such that $\sum_{S \in \mathcal{S}} \prod_{x \in S} \mathbf{w}(x) \geq \frac{\epsilon^k}{k2^k n^{k-1}}$. Note that for each violated variable set $S \in \mathcal{S}$, the probability that we do not find S is $1 - \prod_{x \in S} (q \cdot \mathbf{w}(x)) = 1 - q^k \prod_{x \in S} \mathbf{w}(x)$. Thus, because violated sets in \mathcal{S} are disjoint, the probability that we do not find any violated variable set is at most

$$\prod_{S \in \mathcal{S}} \left(1 - q^k \prod_{x \in S} \mathbf{w}(x)\right) \leq \prod_{S \in \mathcal{S}} \exp\left(-q^k \prod_{x \in S} \mathbf{w}(x)\right) = \exp\left(-q^k \sum_{S \in \mathcal{S}} \prod_{x \in S} \mathbf{w}(x)\right) \leq \exp\left(-q^k \frac{\epsilon^k}{k2^k n^{k-1}}\right)$$

If we choose the constant hidden in q large enough, the probability above is bounded by $\frac{1}{100}$. Thus, with probability at least $\frac{99}{100}$, we reject the instance. \square

6 Non Sublinear-Query Testability

Let \mathbf{A} be a relational structure that does not have, for any $k \geq 2$, a $(k+1)$ -ary near unanimity polymorphism. In this section, we show that $\exists\text{CSP}(\mathbf{A})$ is not sublinear-query testable with one-sided error.

Using Barto's proof of Zádori's Conjecture [2], we know that a finite relational structure \mathbf{A} has a $(k+1)$ -ary near unanimity polymorphism for some $k \geq 2$ if and only if the variety $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is congruence distributive. Furthermore, this condition is equivalent to $\mathcal{V}(\text{Alg}(\mathbf{A}))$ being congruence meet semidistributive and congruence modular [19].

From the argument in Section 4 we know that if $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is not congruence meet semidistributive, then testing $\text{CSP}(\mathbf{A})$ and hence $\exists\text{CSP}(\mathbf{A})$ requires a linear number of queries. Hence, to establish the main result of this section, Theorem 6.6, it suffices to show that $\exists\text{CSP}(\mathbf{A})$ is not sublinear-query testable with one-sided error when $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is not congruence modular. The results in this section make use of ideas developed in [8] and [14].

A *lattice* is an algebra $(L; \wedge, \vee)$ where L is a domain, and each of the operations \wedge, \vee is idempotent, commutative, and associative; and, the absorption law $a \wedge (a \vee b) = a \vee (a \wedge b) = a$ holds. A lattice naturally induces a partial order \leq defined by $a \leq b$ if and only if $a \wedge b = a$. A lattice is *distributive* if it satisfies the identity $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$. A lattice is *finite* if its domain is finite, and is *non-trivial* if its domain has size strictly larger than 1. It is known that a finite lattice has a *bottom element* \perp and a *top element* \top such that for each element a , it holds that $\perp \leq a \leq \top$.

Let $\mathbb{L} = (L; \wedge, \vee)$ be a finite lattice and let $D > 1$ be a constant. We define $\text{Lattice-Eval}(\mathbb{L}, D)$ to be the assignment problem where an instance consists of the following:

- A circuit, on a variable set V , over basis $\{\wedge, \vee\}$ of depth less than $D + D \log_2 |V|$. Here, \wedge and \vee are always assumed to have fan-in 2.
- An element $\ell \in L$.
- A weight function $w : V \rightarrow [0, 1]$.

The assignments associated to an instance are the mappings from V to L ; such a mapping $f : V \rightarrow L$ is considered to be *satisfying* if $C(f) \geq \ell$. Here and in general, when C is a circuit on variable set V and f is an assignment defined on V , we use $C(f)$ to denote the result of evaluating C under f .

Lemma 6.1. *There exists a constant $D > 1$ such that testing $\text{Lattice-Eval}((\{0, 1\}, \{\wedge, \vee\}), D)$ (where $\ell = 1$) with one-sided error requires a linear number of queries.*

Proof. We can view the stated problem as that of testing assignments to logarithmic depth monotone circuits. It is known [32] that there are such circuits for the *majority* function. Hence, it suffices to argue that testing the (uniformly weighted) majority function with one-sided error requires a linear number of queries.

To see this, let us think about the behavior of a one-sided error tester T , given the all-zero assignment $f : V \rightarrow \{0, 1\}$. Notice that f is $1/2$ -far from satisfying the majority function. Hence, when $\epsilon < 1/2$, the tester T must reject f with probability at least $2/3$.

Suppose that T has queried variables in $S \subseteq V$ with $|S| < n/2$. Then, the assignment f' with $f'(x) = 0$ for every $x \in S$ and $f'(x) = 1$ for every $x \in V \setminus S$ is consistent with what T has seen. However, f' satisfies the majority function, and hence T cannot reject f . This means that the query complexity of T must be $\Omega(n)$. \square

Lemma 6.2. *Let \mathbb{L} be a non-trivial finite lattice. For each constant $D > 1$, there exists a constant $D' > 1$ such that there is a linear reduction from the problem $\text{Lattice-Eval}(\{0, 1\}; \wedge, \vee, D)$ (where $\ell = 1$) to the problem $\text{Lattice-Eval}(\mathbb{L}, D')$.*

Proof. We first consider the case where L is a distributive lattice. It is well-known and straightforward to verify that each finite distributive lattice embeds into a finite power of the two-element lattice. We thus view L as a sublattice of a finite power of the two-element lattice, and in particular assume that the domain of L is a subset of $\{0, 1\}^k$. We may further assume that the bottom element of L is $(0, \dots, 0)$.

Let $\mathcal{I} = (C, \ell = 1, \mathbf{w})$ be an instance of the problem $\text{Lattice-Eval}(\{0, 1\}; \wedge, \vee, D)$. Fix $t \in \{0, 1\}^k$ to be an element of L such that $t \neq (0, \dots, 0)$. The instance created is $\mathcal{I}' = (C, t, \mathbf{w})$. An assignment $f : V \rightarrow \{0, 1\}$ is mapped to the assignment $f' : V \rightarrow \{0, 1\}^k$ defined by $f'(x) = (0, \dots, 0)$ if $f(x) = 0$, and $f'(x) = t$ if $f(x) = 1$. It is straightforward to verify that $C(f')$ is equal to 0 or t depending on whether or not $C(f)$ is equal to 0 or 1, respectively. Hence, if f is a satisfying assignment of \mathcal{I} , then f' is a satisfying assignment of \mathcal{I}' .

We claim that if g' is a satisfying assignment of \mathcal{I}' such that $\text{dist}_{\mathcal{I}'}(f', g') < \epsilon$, then there exists a satisfying assignment g of \mathcal{I} such that $\text{dist}_{\mathcal{I}}(f, g) < \epsilon$. This implies that the constant c_1 in the definition of reduction can be taken as $c_1 = 1$. In particular, define $g : V \rightarrow \{0, 1\}$ so that $g(x) = 0$ if $g'(x) = (0, \dots, 0)$, and $g(x) = 1$ otherwise. The inequality $\text{dist}_{\mathcal{I}}(f, g) < \epsilon$ holds as a consequence of the fact that (for each $x \in V$) $f(x) \neq g(x)$ implies $f'(x) \neq g'(x)$; this fact can be verified by a case analysis of the possible values $(0, 1), (1, 0)$ for $(f(x), g(x))$. Now fix r to be an index such that the r th entry t_r of the tuple t is equal to 1. (We extend this subscript notation to assignments mapping to L in the natural fashion.) Since g' is a satisfying assignment, $C(g') \geq t$, implying that $C(g'_r) = C(g')_r \geq t_r = 1$; since $g \geq g'_r$ (by definition of g), $C(g) = 1$ and the assignment g is satisfying, with respect to \mathcal{I} .

We now consider the case where L is a non-distributive lattice. Define $s(x, y, z)$ to be $(x \wedge y) \vee (x \wedge z)$ and define $s'(x, y, z)$ to be $x \wedge (y \vee z)$. Under any assignment to the variables $\{x, y, z\}$, it holds that $s \leq s'$. Fix values $a, b, c \in L$ such that $s(a, b, c) \neq s'(a, b, c)$; such values exist by the assumption that L is non-distributive. For any elements $d, d' \in L$, define $[d, d']$ as $\{e \mid d \leq e \leq d'\}$. It is straightforward to verify that $[s(a, b, c), s'(a, b, c)] \neq L$. Define \mathbb{L}^- to be the sublattice of \mathbb{L} with domain $[s(a, b, c), s'(a, b, c)]$.

By induction, it suffices to show that, for each $D > 1$, there exists $D' > 1$ such that there is a reduction from $\text{Lattice-Eval}(\mathbb{L}^-, D)$ to $\text{Lattice-Eval}(\mathbb{L}, D')$. Let $\mathcal{I}^- = (C^-(x_1, \dots, x_n), \ell, \mathbf{w}^-)$ be an instance of $\text{Lattice-Eval}(\mathbb{L}^-, D)$ where we use $V = \{x_1, \dots, x_n\}$ to denote the variable set of C^- , and let $f^- : V \rightarrow L^-$ be an assignment of \mathcal{I}^- . The instance \mathcal{I} produced by the reduction is (C, ℓ, \mathbf{w}) , where $C(z_1, z_2, z_3, x_1, \dots, x_n)$ is the circuit on variable set $V \cup \{z_1, z_2, z_3\}$ defined as $C^-(x_1^*, \dots, x_n^*)$; each x_i^* is equal to the circuit $(x_i \vee s(z_1, z_2, z_3)) \wedge s'(z_1, z_2, z_3)$. Observe that the depth of the created circuit C is equal to that of C^- , plus a constant. The weight function \mathbf{w} is defined as $\mathbf{w}(z_1) = \mathbf{w}(z_2) = \mathbf{w}(z_3) = 1/4$ and $\mathbf{w}(x) = \mathbf{w}^-(x)/4$ for each $x \in V$. The assignment $f : V \cup \{z_1, z_2, z_3\} \rightarrow L$ produced by the reduction is the extension of f^- that maps (z_1, z_2, z_3) to (a, b, c) .

Clearly, if f^- is a satisfying assignment of \mathcal{I}^- , then f is a satisfying assignment of \mathcal{I} , since $C^-(f^-) = C(f)$. We claim that the constant c_1 in the definition of reduction can be taken as $1/4$. Suppose that g is a satisfying assignment of \mathcal{I} such that $\text{dist}_{\mathcal{I}}(f, g) < \epsilon/4$. Then g must be equal to f on $\{z_1, z_2, z_3\}$, since each of those variables has weight $1/4$. Define $g^-(x) = (g(x) \vee s(a, b, c)) \wedge s'(a, b, c)$, for each $x \in V$; by definition of C , it holds that g^- is a satisfying assignment of \mathcal{I}^- .

Observe that $f(x) = g(x)$ implies that $f^-(x) = g^-(x)$, since $f^-(x) = (f(x) \vee s(a, b, c)) \wedge s'(a, b, c)$. We conclude that $\text{dist}_{\mathcal{L}}(f^-, g^-) < \epsilon$, establishing the claim. \square

Let A be a set. Recall that for binary relations θ and θ' on A , we use $\theta \circ \theta'$ to denote their relational product. We use $\text{Eq}(A)$ to denote the lattice of equivalence relations on A , and we use $0_A = \{(a, a) \mid a \in A\}$ and $1_A = A^2$ to denote the bottom and top elements of $\text{Eq}(A)$, respectively. We define a *pentagon* to be a finite relational structure \mathbf{P} over the signature (or relational structure language) $\{\alpha, \beta, \gamma\}$ containing three binary relation symbols such that $\alpha^{\mathbf{P}}$, $\beta^{\mathbf{P}}$, and $\gamma^{\mathbf{P}}$ are equivalence relations on P , and the following conditions hold in $\text{Eq}(P)$: $\alpha^{\mathbf{P}} \leq \beta^{\mathbf{P}}$, $\beta^{\mathbf{P}} \wedge \gamma^{\mathbf{P}} = 0_P$, $\beta^{\mathbf{P}} \circ \gamma^{\mathbf{P}} = 1_P$, and $\alpha^{\mathbf{P}} \vee \gamma^{\mathbf{P}} = 1_P$. The domain P of a pentagon \mathbf{P} can be naturally decomposed as a direct product $P = B \times C$ in such a way that $\beta^{\mathbf{P}}$ and $\gamma^{\mathbf{P}}$ are the kernels of the projections of P onto B and C , respectively. Then, via the equivalence relation $\alpha^{\mathbf{P}}$, each element $b \in B$ induces an equivalence relation $\alpha_b^{\mathbf{P}} = \{(c, c') \in C \times C \mid ((b, c), (b, c')) \in \alpha^{\mathbf{P}}\}$ on C . For each pentagon \mathbf{P} , we define $\mathbb{L}(\mathbf{P})$ to be the lattice which is the sublattice of $\text{Eq}(C)$ generated by the equivalence relations $\alpha_b^{\mathbf{P}}$ (over $b \in B$).

To each pentagon \mathbf{P} , we associate a 2-sorted relational structure, denoted by \mathbf{P}_2 , which has $B_{\mathbf{P}}$ and $C_{\mathbf{P}}$ as first and second domain, respectively; here, $B_{\mathbf{P}}$ and $C_{\mathbf{P}}$ denote the sets in the decomposition of the domain P as described above. The structure \mathbf{P}_2 is defined on signature $\{R\}$ and has $R^{\mathbf{P}_2} = \{(b, c, c') \in B_{\mathbf{P}} \times C_{\mathbf{P}} \times C_{\mathbf{P}} \mid (c, c') \in \alpha_b^{\mathbf{P}}\}$. The definition of \mathbf{P}_2 comes from [8]. In forming conjunctive queries over this signature $\{R\}$ each variable has a sort (first or second) associated with each variable; an atom $R(x, y, y')$ may be formed if x is of the first sort and y and y' are of the second sort.

When \mathbf{P} is a pentagon, we define $\text{Pent-Eval}(\mathbf{P})$ to be the assignment problem where an instance consists of the following:

- A pp-formula $\phi(X, Y)$ on the signature $\{R\}$ of \mathbf{P}_2 , where the variables in the sets X and Y are of the first and second sort, respectively. (We assume $X \cap Y = \emptyset$.)
- A weight function $\mathbf{w} : X \cup Y \rightarrow [0, 1]$.

The assignments associated to an instance are the mappings $h = h_1 \cup h_2$ where h_1 is a mapping from X to $B_{\mathbf{P}}$ and h_2 is a mapping from Y to $C_{\mathbf{P}}$. Such a mapping is satisfying if it causes ϕ to evaluate to true over \mathbf{P}_2 .

Lemma 6.3. *Let \mathbf{P} be a pentagon. For each $D \geq 1$, there exists a linear reduction from the problem $\text{Lattice-Eval}(\mathbb{L}(\mathbf{P}), D)$ to $\text{Pent-Eval}(\mathbf{P})$.*

Proof. We first observe that there exists a constant E such that $\text{Lattice-Eval}(\mathbb{L}(\mathbf{P}), D)$ linearly reduces to the special case of $\text{Lattice-Eval}(\mathbb{L}(\mathbf{P}), D + E)$ where the assignment must map to the set of generators $G = \{\alpha_b^{\mathbf{P}} \mid b \in B_{\mathbf{P}}\}$ of $L(\mathbf{P})$. Let $s(x_1, \dots, x_q)$ be a fixed lattice term that maps G^q surjectively onto $L(\mathbf{P})$. The reduction, on $(T(x_1, \dots, x_m), \ell, \mathbf{w}), f$, produces

$$(T(s(x_1^1, \dots, x_1^q), \dots, s(x_m^1, \dots, x_m^q)), \ell, \mathbf{w}'), f'.$$

Here, the assignment f' is defined so that, for each $i \in [m]$, it holds that $s(f'(x_i^1), \dots, f'(x_i^q)) = f(x_i)$; the function \mathbf{w}' is defined by $\mathbf{w}'(x_i^j) = w(x_i)/q$ for all $i \in [m]$, $j \in [q]$. It is straightforward to verify that the constant c_1 in the definition of reduction can be taken as $1/q$; the key point is that, in order to modify a value $s(x_i^1, \dots, x_i^q)$ of the new instance, which corresponds to the input

x_i to T , it is necessary to change at least one of the values x_i^1, \dots, x_i^q , whose weight is $1/q$ times the weight $w(x_i)$.

It thus suffices to give a linear reduction from this special case of $\text{Lattice-Eval}(L(\mathbf{P}), D + E)$ to $\text{Pent-Eval}(\mathbf{P})$, which is what we now do. Let $(T(x_1, \dots, x_m), \ell, \mathbf{w}), f$ be an input to the first problem. We make use of a construction in the literature (introduced in [8] and also employed in [14]) which allows us to create, from the circuit $T(x_1, \dots, x_m)$, a pp-formula $\phi_T(x_1, \dots, x_m, y_1, y_2)$ (over \mathbf{P}_2) such that, for all $b_1, \dots, b_m \in B_{\mathbf{P}}$ and all $c_1, c_2 \in C_{\mathbf{P}}$, we have that $\phi_T(b_1, \dots, b_m, c_1, c_2)$ holds on \mathbf{P}_2 if and only if (c_1, c_2) is in the equivalence relation given by $T(\alpha_{b_1}^{\mathbf{P}}, \dots, \alpha_{b_m}^{\mathbf{P}})$, where here it is understood that T is evaluated in the lattice $\mathbb{L}(\mathbf{P})$. For the sake of completeness, we briefly specify the version of the construction used here. The construction is defined inductively. When $T = x_i$, we have $\phi_T(x_1, \dots, x_m, y_1, y_2) = R(x_i, y_1, y_2)$. When $T = T_1 \wedge T_2$, we have $\phi_{T_1}(x_1, \dots, x_m, y_1, y_2) \wedge \phi_{T_2}(x_1, \dots, x_m, y_1, y_2)$. When $T = T_1 \vee T_2$, set $u = |C_{\mathbf{P}}|$. Let $z_{0,2}$ and $z_{i,j}$, where $i = 1, \dots, u$ and $j = 1, 2$, be variables of the second sort, and identify $y = z_{0,2}$ and $y' = z_{u,2}$. Then ϕ_T is the formula $\exists z_{1,1} \exists z_{1,2} \dots \exists z_{u-1,1} \exists z_{u-1,2} \exists z_{u,1} \bigwedge_{i=1}^u (\phi_{T_1}(x_1, \dots, x_m, z_{i-1,2}, z_{i,1}) \wedge \phi_{T_2}(x_1, \dots, x_m, z_{i,1}, z_{i,2}))$, where here all of the variables z_{\cdot} are existentially quantified, other than y and y' .

The reduction produces the input $(\bigwedge_{i \in [u^2]} \phi_T(x_1, \dots, x_m, y_i^1, y_i^2), \mathbf{w}'), f'$, where $u = |C_{\mathbf{P}}|$, and f' and \mathbf{w}' are described as follows. Define $f'(x_i)$ so that $f(x_i) = \alpha_{f'(x_i)}^{\mathbf{P}}$, and define f' on the (y_i^1, y_i^2) so that $\ell = \{(f'(y_i^1), f'(y_i^2)) \mid i \in [u^2]\}$. Define $\mathbf{w}'(y_i^1) = \mathbf{w}'(y_i^2) = 1/(2u^2 + 1)$, and define $\mathbf{w}'(x_i) = \mathbf{w}(x_i)/(2u^2 + 1)$. The reduction works with $c_1 = 1/(2u^2 + 1)$, for if f' is within distance $c_1\epsilon$ of a satisfying assignment, then the satisfying assignment must be equal on the variables (y_i^1, y_i^2) . \square

Lemma 6.4 ([8]). *Let \mathbf{B} be a finite relational structure such that $\mathcal{V}(\text{Alg}(\mathbf{B}))$ is not congruence modular. There exists a relational structure \mathbf{A} defined on a signature including three binary relation symbols α, β , and γ which are preserved by the operations of some finite algebra in $\mathcal{V}(\text{Alg}(\mathbf{B}))$, such that the following hold:*

- *There exists a finite set \mathcal{P} of pentagons where for each $\mathbf{P} \in \mathcal{P}$, the domain P of \mathbf{P} is a subset of A , and it holds that $\alpha^{\mathbf{P}} = \alpha^{\mathbf{A}} \cap P^2$, $\beta^{\mathbf{P}} = \beta^{\mathbf{A}} \cap P^2$, and $\gamma^{\mathbf{P}} = \gamma^{\mathbf{A}} \cap P^2$. Moreover, there exists $\mathbf{P} \in \mathcal{P}$ such that $\mathbb{L}(\mathbf{P})$ is a non-trivial lattice.*
- *For each $k \geq 1$, there exists a relation $D_k \subseteq A^k$ which is pp-definable over \mathbf{A} such that for any elements $a_1, \dots, a_k \in A$, the tuple (a_1, \dots, a_k) is in D_k if and only if there exists a $\mathbf{P} \in \mathcal{P}$ such that all of the elements a_1, \dots, a_k are contained in the domain P of \mathbf{P} .*

Theorem 6.5. *Let \mathbf{A} be a relational structure satisfying the conditions described in Lemma 6.4, and let $\mathbf{P} \in \mathcal{P}$ be a non-trivial pentagon whose domain is not contained in that of any other pentagon in \mathcal{P} . There exists a linear reduction from $\text{Pent-Eval}(\mathbf{P})$ to $\exists\text{CSP}(\mathbf{A})$.*

Note that this theorem makes use of a construction from [8, Theorem 7] and shares elements in common with the proof of [14, Theorem 13].

Proof. Let $(\phi(x_1, \dots, x_m, y_1, \dots, y_\ell), \mathbf{w}), h$ denote the input to the first problem. Let $x_{m+1}, \dots, x_{m'}$ denote the quantified variables of the first sort in ϕ , and let $y_{\ell+1}, \dots, y_{\ell'}$ denote the quantified variables of the second sort in ϕ . We use the translation of [8, Theorem 7] to obtain a formula $\phi'(x'_1, \dots, x'_{m'}, y'_1, \dots, y'_{\ell'}, v'_1, \dots, v'_{|P|})$, but instead of adding the conjunct of the form $\Delta(\dots)$, we add the conjunct $D_{m'+\ell'+|P|}(x'_1, \dots, x'_{m'}, y'_1, \dots, y'_{\ell'}, v'_1, \dots, v'_{|P|})$ where $|P|$ denotes the size of the

domain P of \mathbf{P} and the v_i are fresh variables. Let b^* be a fixed element of $B_{\mathbf{P}}$, and let c^* be a fixed element of $C_{\mathbf{P}}$. Define the assignment h' as follows: $h'(x'_i) = (h(x_i), c^*)$ for each x_i , $h'(y'_i) = (b^*, h(y_i))$ for each y_i , and let $h(v'_1), \dots, h(v'_{|P|})$ be an enumeration of the elements of P . Set \mathbf{w}' so that $\mathbf{w}'(v'_i) = 1/(|P| + 1)$ for each variable v'_i , and so that $\mathbf{w}'(u) = \mathbf{w}(u)/(|P| + 1)$ for each other variable u of ϕ' . It is straightforward to verify that the reduction that outputs $(\phi', \mathbf{w}'), h'$ is correct. \square

Combining Lemmas 6.1, 6.2, and 6.3 and Theorem 6.5, using Lemma 2.4, we get the following.

Theorem 6.6. *Let \mathbf{A} be a relational structure such that $\mathcal{V}(\text{Alg}(\mathbf{A}))$ is not congruence modular. Then, $\exists\text{CSP}(\mathbf{A})$ is not sublinear-query testable with one-sided error.*

From the argument at the beginning of this section, we obtain the following as a corollary.

Theorem 6.7. *Let \mathbf{A} be a relational structure that has not $(k+1)$ -ary near unanimity polymorphism for any $k \geq 2$. Then, $\exists\text{CSP}(\mathbf{A})$ is not sublinear-query testable with one-sided error.*

References

- [1] K. A. Baker and A. F. Pixley. Polynomial interpolation and the chinese remainder theorem for algebraic systems. *Mathematische Zeitschrift*, 143(2):165–174, 1975.
- [2] L. Barto. Finitely related algebras in congruence distributive varieties have near unanimity terms. *Canad. J. Math.*, 65(1):3–21, 2013.
- [3] L. Barto. The collapse of the bounded width hierarchy. *Journal of Logic and Computation*, 2014.
- [4] L. Barto. The constraint satisfaction problem and universal algebra. *The Bulletin of Symbolic Logic*, 21:319–337, 9 2015.
- [5] L. Barto and M. Kozik. Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM*, 61(1), 2014.
- [6] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1), 2005.
- [7] A. Bhattacharyya and Y. Yoshida. An algebraic characterization of testable Boolean CSPs. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 123–134, 2013.
- [8] S. Bova, H. Chen, and M. Valeriote. Generic expression hardness results for primitive positive formula comparison. *Inf. Comput.*, 222:108–120, 2013.
- [9] A. Bulatov and D. Marx. The complexity of global cardinality constraints. *Logical Methods in Computer Science*, 6:1–27, 2010.
- [10] A. A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.*, 12(4):24, 2011.

- [11] A. A. Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34, 2013.
- [12] S. Burris and H. P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, New York-Berlin, 1981.
- [13] S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer. Approximate constraint satisfaction requires large LP relaxations. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 350–359, 2013.
- [14] H. Chen and M. Valeriote. Learnability of solutions to conjunctive queries: The full dichotomy. In *Proceedings of The th Conference on Learning Theory*, pages 326–337, 2015.
- [15] M. E. Dyer and D. Richerby. An effective dichotomy for the counting constraint satisfaction problem. *SIAM J. Comput.*, 42(3):1245–1274, 2013.
- [16] T. Feder and M. Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
- [17] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 474–483, 2002.
- [18] O. Goldreich, editor. *Property Testing*, volume 6390 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [19] D. Hobby and R. McKenzie. *The structure of finite algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, RI, 1988. Revised edition: 1996.
- [20] K. Idziak and P. M. Idziak. Decidability problem for finite Heyting algebras. *J. Symbolic Logic*, 53(3):729–735, 1988.
- [21] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.*, 39(7):3023–3037, 2010.
- [22] P. Jeavons, D. Cohen, and M. C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.
- [23] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 767–775, 2002.
- [24] M. Kozik, A. Krokhin, M. Valeriote, and R. Willard. Characterizations of several Maltsev conditions. *Algebra Universalis*, 73(3-4):205–224, 2015.
- [25] I. Newman. Property testing of massively parametrized problems - a survey. *Property Testing*, 6390(Chapter 8):142–157, 2010.
- [26] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 245–254, 2008.

- [27] D. Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends® in Theoretical Computer Science*, 5:73–205, 2010.
- [28] R. Rubinfeld and A. Shapira. Sublinear time algorithms. *SIAM Journal on Discrete Mathematics*, 25(4):1562–1588, 2011.
- [29] J. Thapper and S. Zivny. The power of linear programming for valued CSPs. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 669–678, 2012.
- [30] J. Thapper and S. Zivny. The complexity of finite-valued CSPs. In *Symposium on Theory of Computing Conference, STOC’13, Palo Alto, CA, USA, June 1-4, 2013*, pages 695–704, 2013.
- [31] J. Thapper and S. Zivny. Sherali-adams relaxations for valued CSPs. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 1058–1069, 2015.
- [32] L. G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5(3):363–366, 1984.
- [33] Y. Yoshida. Testing list H -homomorphisms. *Computational complexity*, pages 1–37, 2014.